Chasing the Tail with Domain Generalization: A Case Study on Frequency-Enriched Datasets

Manoj Kumar[1], **Anna Rumshisky**[1,2], **Rahul Gupta**[1]
[1]Alexa AI, Amazon
[2]Department of Computer Science, University of Massachusetts Lowell
{abithm,gupra}@amazon.com
arum@cs.uml.edu

Abstract

Natural language understanding (NLU) tasks are typically defined by creating an annotated dataset in which each utterance is encountered once. Such data does not resemble real-world natural language interactions in which certain utterances are encountered frequently, others rarely. For deployed NLU systems, this is a vital problem, since the underlying machine learning (ML) models are often fine-tuned on typical NLU data, in which utterance frequency is never factored in, and then applied to real-world data with a very different distribution. Such systems need to maintain interpretation consistency for the high-frequency (head) utterances, while also doing well on low-frequency (tail) utterances. We propose an alternative strategy that explicitly uses utterance frequency in training data to learn models that are more robust to unknown distributions. We present a methodology to simulate utterance usage in two public corpora and create two new corpora with head, body and tail segments. We evaluate several methods for joint intent classification and named entity recognition (referred to as IC-NER), and propose to use two domain generalization (DG) approaches that we adapt to sequence labeling task. The DG approaches demonstrate up to 7.02% relative improvement in semantic accuracy over baselines on the tail data. We provide insights as to why the proposed approaches work and show that the reasons for the observed improvements do not align with those reported in previous work.

1 Introduction

In academic research, natural language understanding (NLU) tasks are typically defined by creating annotated data, and then that data is used to train and evaluate machine learning models designed to solve that task. In such datasets, each utterance is typically encountered only once. But real-world natural language interactions do not look like that – in the real world, frequency matters. When people interact with each other "in the wild", some things are said often ("Time to go to bed!"), others are infrequent to the point of being unique.

The same holds for how people interact with digital assistants such as Alexa, Siri, or Google Assistant, which we use as the case study in this paper. The backbone of such commercial systems is the task of joint intent classification and named entity recognition (IC-NER) (Su et al., 2018; Coucke et al., 2018; Anantha et al., 2021). The goal of this task is to identify the intended action (play music, open calendar, etc) and actionable slots (names, places, objects, etc) from a user utterance.

The underlying joint IC-NER models must correctly handle both the frequently occurring requests and a long tail of less common entities. But in the common IC-NER corpora such as SNIPS (Coucke et al., 2018), there is no way to distinguish between requests for generic entities ("*play music from youtube*") and requests for a low-frequency entity ("*help me locate a game called the master of ballantrae*"). IC-NER models are fine-tuned on all training data, and then applied to real-world data with a very different distribution.

In order to mitigate this issue, this work proposes a method for creating annotated data which explicitly factors in utterance frequency. We divide an NLU dataset into three disjoint segments: head (most frequent utterances), tail (least frequent utterances) and body (all remaining utterances). In this work, we define a *segment* as a subset of the dataset with similar characteristics, for example the head segment contains utterances with high frequencies in the real world. We then develop learning strategies which benefit from the token and label distributions in the head, body, and tail segments of the resulting frequency-enriched datasets.

We simulate utterance usage patterns using two common public corpora for the IC-NER task: SNIPS (Coucke et al., 2018) which con-

Table 1: Selected examples from head and tail segments in the newly created corpora: SNIPSesv and TOPesv. Utterances from head segments include the repetition counts. Tokens with slot labels are **boldfaced**.

	SNIPSesv	TOPesv
Head	"play music off **youtube**": 76 " play some **google music**": 36	"is **the weather** causing traffic delays **today**": 65 "where is **macys**": 46 "what **new movies** start **this weekend**": 32
Tail	"add **outside the dream syndicate** to **millicent's fresh electronic** playlist" "what s the weather in **south punta gorda heights**" "add **9th inning** to **my bossa nova dinner** playlist"	"what is the quickest route to get to **valdosta** from **atlanta**" "how long does it take to **drive** from **adair** to **chelsea**"

tains real-world utterances directed towards the SNIPS voice assistant, and the Facebook Dialog Corpus (TOP; Gupta et al. 2018) which is a crowd-sourced collection of natural language queries related to navigation and event inquiries, creating two frequency-enriched datasets (SNIPSesv and TOPesv). Our methodology is based on entity search volumes, which allows us to emulate a realistic utterance frequency distribution in the data. Utterances are then upsampled according to their estimated frequency. SNIPSesv and TOPesv datasets separate test data for head, body and tail segments, enabling the comparison of model performance on each segment. The proposed methodology can be easily extended to other NLU tasks such as part-of-speech tagging, sentence generation, or question answering.

Using our frequency-enriched datasets, we compare IC-NER performance of several methods. We propose modifications to two domain generalization (DG; (Blanchard et al., 2011)) approaches: domain masks for generalization (DMG; Chattopadhyay et al. 2020) and optimal transport (OT; (Zhou et al., 2020a)). We adapt these methods for IC-NER and demonstrate up to 7.02% relative improvement in semantic accuracy on the tail data over strong baselines.

We provide insights as to why the proposed DG approaches work, showing that OT learns segment-invariant representations using segment classification analysis. Our analysis using random-valued masks reveals that performance improvements by DMG are rather likely due to the training process resembling an enhanced version of dropout, rather than learning segment-specific mask parameters, an observation which does not align with those reported in previous work. We corroborate our observations in NLU with similar findings on a related task from computer vision, for which DMG was originally proposed.

The main contributions of this work are thus as follows: (i) We simulate utterance usage frequency for two public NLU corpora. To the best of our knowledge, these frequency-enriched datasets are the first attempt to explicitly incorporate utterance usage information in NLU. (ii) We adapt two domain generalization approaches to the sequence labeling task in NLU and show improvement over strong baselines on the tail segment, using the frequency-enriched data. (iii) We demonstrate that the reasons for improved performance from DMG do not align with those reported in previous work.

2 Background

2.1 Improving tail recognition

Previous work on head to tail transfer learning has typically focused on assigning classes to either head or tail based on the number of examples present in each class (Xiao et al., 2021; Raunak et al., 2020). Our problem setting is different in that we divide the dataset into head, body and tail based on the estimated usage frequency of each utterance. For example, in our case, the utterances belonging to a common class (such as "play music" intent) may not all be assigned to the head segment, but rather may be split between head, body, and tail, depending on their frequencies.

Since our problem setting presumes a different definition of head and tail, many of the methods (Kang et al., 2020; Ouyang et al., 2016; Cao et al., 2019) developed for head-to-tail transfer are not directly applicable in our case.

2.2 Domain generalization approaches

Domain generalization techniques (Blanchard et al., 2011) are a subset of transfer learning approaches where multiple *domains* with different label distributions and class-conditional distributions are used for model building. As distinct from domain adaption, no data from the target domain(s) is assumed available for training/adaptation. We wanted to investigate DG methods for our scenario, since this would allow us to treat head, tail, and body segments as virtual domains, without making any

specific assumptions about the data and label distributions in each segment.

A variety of DG approaches have been proposed: kernel-based optimization methods (Blanchard et al., 2011, 2021; Muandet et al., 2013), augmenting with synthetic data perturbed using loss gradients (Shankar et al., 2018), learning a transformation to jointly classify domains and labels (Zhou et al., 2020b), learning a segment-invariant feature space by minimizing the optimal transport between domain pairs (Zhou et al., 2020a), etc. Broadly, these approaches learn to project datapoints from different segments into equivalent feature spaces for data representation, which improves performance. This paradigm closely resembles meta-learning, with the difference being that meta-learning assumes access to labeled samples from the target segment during the meta-testing phase (Ravi and Larochelle, 2017). An alternative set of approaches focuses on learning segment-specific knowledge, e.g., using outputs from a model trained on seen segments to train a model for unseen segments (Zhou et al., 2021) or selecting convolution activations to create segment-specific subnetworks in the model (Chattopadhyay et al., 2020; Mallya et al., 2018; Berriel et al., 2019).

DG has been relatively less explored in NLU when compared to computer vision. A handful of works have applied DG for semantic parsing: Wang et al. (2021) employed an adaptation of MAML (Finn et al., 2017) to simulate new segments, Marzinotto et al. (2019) used an adversarial domain classifier as a regularization technique. We adapt two categories of DG approaches: learning representations which are segment-specific (DMG; Chattopadhyay et al. 2020) and segment-invariant (optimal transport; (Zhou et al., 2020a)). We apply these approaches for generalizing IC-NER performance from head, body and tail segments.

3 Methods

3.1 Dataset preparation

Both SNIPS (Coucke et al., 2018) and TOP (Gupta et al., 2018) contain almost exclusively unique utterances, and SNIPS is purposefully designed to contain a balanced number of utterances per intent. Following Chen et al. (2019), IC-NER models are commonly evaluated on data that excludes nested intents, since BERT-based architectures make handling nested intents challenging. In order to enable fair comparison of model performance, we follow this strategy and remove nested intents from TOP. We also remove all utterances labeled with "Unsupported" intent.

3.1.1 Estimating usage frequency

In order to estimate usage frequency of each utterance, we use the internet search volumes of each labeled entity (defined as a token labeled with a slot, e.g., ArtistName). We hypothesize that the utterance's usage frequency is influenced primarily by the mentioned entities (e.g., *master of ballantrae* in Section 1) and not the remaining tokens (e.g., stop words, *play*, *order*, etc)

We collect the monthly entity search volume (denoted esv) averaged over the last year using the Google AdWords API[1]. We estimate the utterance search volume as mean esv for all entities, assuming that each entity contributes equally to the utterance usage. For example, consider the following utterance in the SNIPS corpora: "*Book reservations at a restaurant in Olton around supper time*". There are two labeled entities in it: Olton (city) and supper (time interval). Monthly search volumes in Google for each entity are 266 and 33.1K respectively. Hence, the estimated utterance usage esv_u is 16.7K. In a similar manner, we estimate the usage frequency of all utterances in SNIPS and TOP.

Another option for estimating usage frequencies is to use utterance perplexity estimated by a high-quality pre-trained language model. In preliminary analysis, we used the perplexities from GPT-2 to approximate usage frequency. We did not find that this method produced good estimates of usage frequencies in spoken requests to digital assistants, likely due to the domain difference of the data used pre-training of GPT-2. Pre-training on in-domain data can be used to address this in the future, potentially enabling this alternative strategy for estimating utterance frequency.

3.1.2 Utterance sampling

We used the frequency estimate for each utterance to determine the upsampling factor for that utterance. Intuitively, an utterance with a higher esv_u should be sampled more, and is more likely to be present in the head segment.

We normalize the obtained search volume to derive a probability distribution p_u over utterances. However, we compared the resulting distribution

[1] https://developers.google.com/adwords/api/

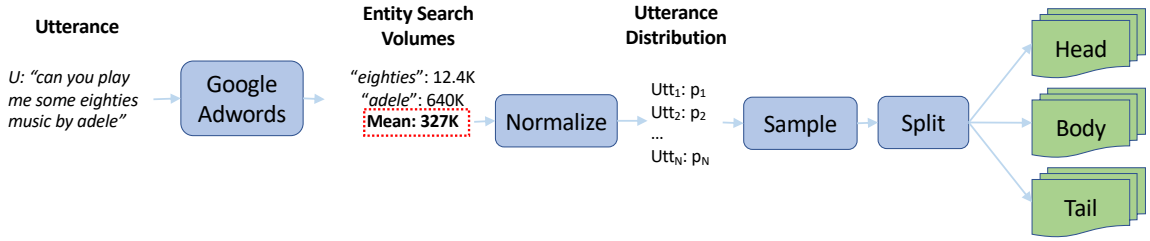

Figure 1: Overview of the dataset preparation process. For each utterance from the original train, dev and test sets from SNIPS and TOP, we estimate the utterance frequency. The frequency is normalized to a probability distribution which is used to sample utterances.

against the utterance in a proprietary commercial dataset[2], and observed that while p_u gave reasonable estimates in many cases, it was not well calibrated. Specifically, it produced a heavy skew in favor of frequent utterances, possibly due to the fact that we were only able to approximate frequencies at the entity, rather than utterance level. Sampling directly from p_u would therefore have produced a corpus with a small number of unique utterances and many repetitions, while omitting most utterances from the original dataset.

To avoid this issue, we cap the maximum sampling probability p_{max} of an utterance. We define p_{max} to be the probability of the most common utterance, defined as follows:

$$p_{max} = \frac{|u_{max}|}{\sum_i |u_i|} \quad (1)$$

where u_i denotes a unique utterance and u_{max} denotes the most common unique utterance. We empirically determine p_{max} = 0.00245 using the proprietary corpus of user queries with semantically similar intent labels to SNIPS and TOP. Further details are provided in the Appendix.

3.1.3 Splitting into head, body and tail

We create frequency-enriched versions of the TOP and SNIPS datasets using the capped probability distribution to sample utterances with replacement. We fix the total number of utterances (N) in the new corpus and sample utterances using the capped distribution until we collect N utterances. We segment the upsampled corpus into head, body, and tail, where head and tail are designed to contain fewer utterances than body. The frequency of utterances in the head and tail segments is very high or very low, respectively. We assign 10% most frequent utterances to head, 10% least frequent utterances to tail and remaining utterances to body[3]. We create the train and test partitions of SNIPSesv and TOPesv separately from the original train and test partitions, hence resulting in six segments (3 train + 3 test) for each corpus.

We report utterance and label statistics of the resulting datasets in Table 2. In both SNIPSesv and TOPesv, the head segment contains relatively fewer unique utterances than other segments, but each unique utterance is repeated multiple times. Note that the head segment does not contain the complete set of labels (intents and classes) found in the original corpora. Specifically, the head segment in SNIPSesv and TOPesv contain only 30.5% and 38.4% of all the slot labels in the original segment, respectively. Some intent labels are also missing in other segments in TOPesv, likely because the TOP corpus (Gupta et al., 2018), unlike SNIPS, has a non-uniform intent distribution. In Table 1, we provide representative examples from head and tail segments in the newly created corpora. Note that utterances with popular/generic entities (e.g., youtube, weather) are likely to end up in the head segment when compared to less widely used entities.

3.2 Domain Generalization Approaches

As the omitted intent statistics in Table 2 suggest, head, body and tail segments of both datasets have very different label distributions $P(Y)$. At the same time, since utterances are sampled according to the entity search volume, each segment has a different distribution over tokens $P(X)$ (Table 1). These differences in label and token distributions motivate our choice of DG approaches for improv-

[2]See Appendix for details.

[3]Utterances are not shared between segments, hence the exact fraction of utterances across head, body and tail may not be equal to 10%-80%-10%

Table 2: Dataset statistics for head, body and tail segments in SNIPSesv and TOPesv, along with the respective original corpora ("Original" segment). Splits (train, dev and test) for each segment are created using the corresponding splits from the original corpora. For each split within a segment, the total utterance count (Utt), unique utterance count (Uniq Utt), average repetition of unique utterances (Rep), and missing labels are provided. The total number of intents and slot labels are provided against the respective column headers.

		SNIPSesv					TOPesv				
Segment	**Split**	**Utt**	**Uniq Utt**	**Rep**	**# Missing Intents (7)**	**# Missing Slots (72)**	**Utt**	**Uniq Utt**	**Rep**	**# Missing Intents (12)**	**# Missing Slots (26)**
Original	Train	13084	12860	1.02	-	-	20265	19764	1.03	-	-
	Dev	700	695	1.01	-	2	2955	2937	1.01	5	1
	Test	700	699	1.00	-	2	5884	5834	1.01	4	2
Head	Train	1323	34	38.91	2	44	1748	40	43.7	6	12
	Dev	73	8	9.13	5	50	253	26	9.73	9	16
	Test	74	11	6.73	1	48	515	40	12.88	7	15
Body	Train	10453	2537	4.12	-	2	13922	5668	2.46	-	1
	Dev	558	230	2.43	-	11	2020	749	2.70	2	5
	Test	557	267	2.09	-	3	4063	1634	2.49	2	5
Tail	Train	1308	1308	1.00	-	2	1740	1740	1.00	3	7
	Dev	69	69	1.00	-	21	252	252	1.00	2	5
	Test	69	69	1.00	-	14	508	508	1.00	3	5

ing performance on unseen segments (Blanchard et al., 2011).

Both DG approaches explored in this work, DMG (Chattopadhyay et al., 2020) and OT (Zhou et al., 2020a), assume that the model can be broken down into a feature extractor F_Ψ and a task network T_Θ. A typical feature extractor and task network for IC-NER are BERT-based pretrained model and sequence/slot classification network respectively (Chen et al., 2019).

3.2.1 Domain Masks for Generalization (DMG)

DMG encodes segment knowledge in *masks* ($\tilde{\mathbf{m}}^d$), which are segment-specific parameters jointly learnt with F_Ψ and T_Θ. For segment d, we extract binary activations m^d from masks as follows:

$$m^d \sim \text{Bernoulli}(\sigma(\tilde{\mathbf{m}}^d)) \quad (2)$$

where σ represents the sigmoid activation function. During forward pass, we multiply each activation by m^d to compute the effective activation passed to the next layer. Hence, masks serve as layer-wise "on"/"off" gates within T_θ. Masks are sampled during training, hence a different set of neurons are activated for different mini-batches within the same segment.

Similar to the original formulation of DMG (Chattopadhyay et al., 2020), we ensure that masks are incentivized to learn segment-specific information and avoid learning similar representations for all segments by using a soft overlap loss (sIoU; Rahman and Wang 2016). The soft-overlap loss is used in place of Jaccard Similarity Coefficient which is non-differentiable and hence cannot be optimized with gradient descent. Specifically, we compute:

$$\text{sIoU}(\tilde{\mathbf{m}}^{d_i}, \tilde{\mathbf{m}}^{d_j}) = \frac{\tilde{\mathbf{m}}^{d_i} \cdot \tilde{\mathbf{m}}^{d_j}}{\sum(\tilde{\mathbf{m}}^{d_i} + \tilde{\mathbf{m}}^{d_j} - \tilde{\mathbf{m}}^{d_i} \odot \tilde{\mathbf{m}}^{d_j})}$$

At each mini-batch, we compute $\text{sIoU}(\tilde{\mathbf{m}}^{d_i}, \tilde{\mathbf{m}}^{d_j})$ for every segment pair and sum across all pairs. This soft-overlap loss is added to the classification loss and used as the overall objective for optimization.

$$\mathcal{L}_{DMG} = \frac{1}{n}\sum_i \mathcal{L}_{class}(\mathbf{x_i}, y_i) + \lambda_{DMG} \sum_{d_i, d_j \in d} \text{sIoU}(\tilde{\mathbf{m}}^{d_i}, \tilde{\mathbf{m}}^{d_j}) \quad (3)$$

where n, d and $\mathcal{L}_{class}$ represent the mini-batch size, set of segments in the mini-batch, and the classification loss function. At test time, we do not have segment labels for a sample. We arrive at the predicted label by computing the mean prediction obtained with all segment-specific masks.

3.2.2 Optimal Transport

Optimal transport learns segment-invariant feature representations by ensuring feature compactness, i.e., samples from the same class across different segments are brought close to each other and vice versa. Assuming $c : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^+$ is the cost

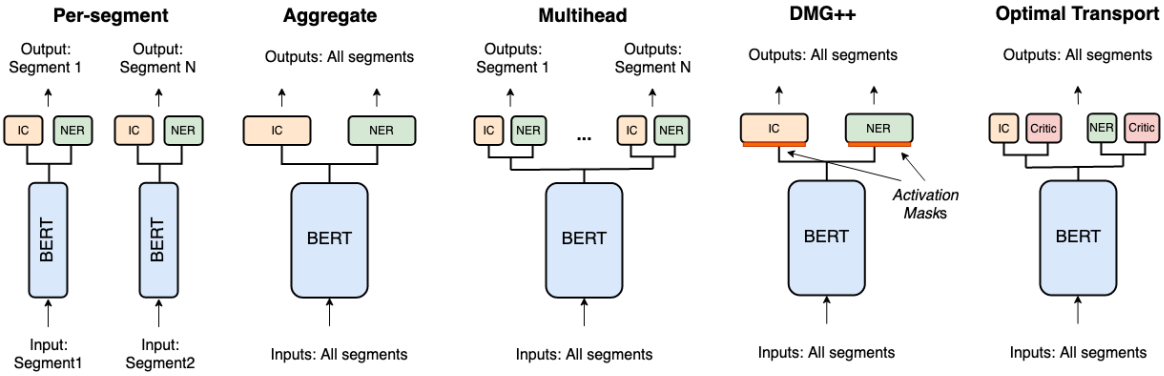

Figure 2: Illustrating the different approaches used in this work: baselines Per-segment, aggregate and multihead, and DG approaches: DMG++ and Optimal Transport.

function for transporting an unit mass from $\mathbf{x}_i$ to $\mathbf{x}_j$, the p-th order Wasserstein distance between d_i and d_j is:

$$W_p^p(d_i, d_j) = \inf_{\gamma \in \Pi(d_i, d_j)} \int_{\mathbb{R}^n \times \mathbb{R}^n} c(\mathbf{x_i}, \mathbf{x_j}) d\gamma(\mathbf{x_i}, \mathbf{x_j}) \tag{4}$$

where $\Pi(d_i, d_j)$ is a collection of all joint probability measures on $\mathbb{R}^n \times \mathbb{R}^n$ with marginals d_i and d_j. Following Zhou et al. (2020a) and from the Kantorovich-Rubinstein theorem (Kantorovich and Rubinshtein, 1958), the first order Wasserstein distance can be given as:

$$W_1(d_i, d_j) = \sup_{\|f\|_L < 1} \mathbb{E}_{x \in d_i} f(x_i) - \mathbb{E}_{x \in d_i} f(x_j) \tag{5}$$

Given sets $X_i = \{\mathbf{x}_i\}_{i=1}^{N_i}$ and $X_j = \{\mathbf{x}_j\}_{j=1}^{N_j}$ from segments d_i and d_j respectively, we can compute the empirical Wasserstein distance between these two sets as:

$$W_1(X_i, X_j) = \frac{1}{N_i} \sum_{\mathbf{x}_i} f(\mathbf{x}_i) - \frac{1}{N_j} \sum_{\mathbf{x}_j} f(\mathbf{x}_j) \tag{6}$$

where f represents a learnable function which transforms inputs to segment-invariant representations. In this work, we parameterize $f = F_\Psi \circ C_\Omega$, where C_Ω is a critic function that is applied on the output from the feature extractor. At each training mini-batch, we compute the critic loss $\mathcal{L}_C$ as the sum of absolute pairwise Wasserstein-1 distances (Eq. 6) between all segment pairs. The critic loss is jointly optimized with the classification loss to learn representations that minimize segment variations while maximizing classification performance.

$$\mathcal{L}_{OT} = \frac{1}{n} \sum_i \mathcal{L}_{class}(\mathbf{x_i}, y_i) + \lambda_{OT} \sum_{d_i, d_j \in d} W_1(X_i, X_j) \tag{7}$$

3.3 Baselines

We compare DG approaches with three baselines: Per-segment, Aggregate and Multihead models. Among these three baselines, we experiment with shared and separate networks for the feature extractor F_Ψ and task networks T_θ (Figure 2). In the per-segment baseline, we construct a separate model for each segment, and train them using respective segment's data. In the multihead baseline, F_Ψ is shared between segments while a different T_Θ is trained for each segment. In the aggregate baseline, both F_Ψ and T_Θ are shared between the segments. For the first two baselines where we have multiple task networks, we predict the intent and slot labels for a test sample by computing the mean prediction from all segment-specific models.

4 Experiments

4.1 Model Components

We use the pretrained BERT-base model (Devlin et al., 2019) as the feature extractor network F_Ψ. The task network T_Θ consists of two sub-networks: (i) The IC network predicts the intent given the CLS token embedding using a single feed-forward layer (ii) The NER network uses a similar feed-forward layer to predict the slot at each word given the hidden state from the last BERT layer. Similar to Chen et al. (2019), we use the hidden state of the first sub-word token of each word for slot prediction. We update parameters of both IC and

Table 3: IC-NER performance on SNIPSesv (top) and TOPesv (bottom) corpora for baselines: Per-segment, Aggregate and Multihead; and domain generalization approaches: DMG++, Optimal Transport and Combined

	Head		Body		Tail		Original	
Approach	**Sem**	**SlotF1**	**Sem**	**SlotF1**	**Sem**	**SlotF1**	**Sem**	**SlotF1**
Per-segment	**87.84**	**96.73**	84.74	94.57	82.61	92.69	83.43	93.64
Aggregate	77.03	95.34	87.97	95.60	81.16	91.81	86.14	94.46
Multihead	**87.84**	**96.73**	85.28	94.75	81.16	91.36	84.43	94.05
DMG++	**87.84**	**96.73**	88.33	95.59	**88.41**	**93.87**	**87.00**	**94.74**
Optimal Transport	77.03	95.34	88.51	95.77	85.51	93.28	86.43	94.26
Combined	77.03	95.34	**89.95**	**96.32**	85.51	93.28	86.29	94.42

	Head		Body		Tail		Original	
Approach	**Sem**	**SlotF1**	**Sem**	**SlotF1**	**Sem**	**SlotF1**	**Sem**	**SlotF1**
Per-segment	88.54	96.93	88.53	95.15	84.06	93.09	86.71	93.49
Aggregate	88.74	97.10	**91.31**	**96.29**	86.22	94.01	88.95	94.67
Multihead	**92.23**	98.27	90.16	95.87	87.40	94.32	88.71	94.51
DMG++	88.93	97.06	90.18	95.94	86.81	93.91	89.03	94.63
Optimal Transport	91.46	**98.71**	91.19	96.25	87.40	**94.60**	**89.34**	**94.88**
Combination	88.54	97.58	90.67	96.01	**87.60**	93.74	88.73	94.40

NER networks using a joint classification loss $\mathcal{L}_{IC}$ + $\mathcal{L}_{NER}$ in order to benefit from any shared knowledge between IC and NER tasks.

4.2 Adapting DMG and OT for NER

Note that the DMG model learns a single mask parameter per segment, i.e it learns one mask for IC ($\tilde{\mathbf{m}}^d_{IC}$) and another mask for NER ($\tilde{\mathbf{m}}^d_{NER}$). This implies that $\tilde{\mathbf{m}}^d_{NER}$ is common across all tokens in the segment and the same activations in F_Ψ are selected for all tokens. This constrains the learning process, since different tokens can benefit from selecting different activations when learning segment-specific representations. To support this, we propose formulating the mask parameters as a function of the segment *and* the token embedding:

$$\tilde{\mathbf{m}}^d_t = w^d h_t \; + \; b^d \tag{8}$$

where h_t represents activation from F_Ψ for token t. We introduce a weight vector w^d and bias b^d for each segment. The masks are sampled using $\tilde{\mathbf{m}}^d_t$ similar to Eq. 2. We refer to this modified version of DMG as DMG++. Similarly, we use two critic networks for OT: $C_{\Omega,IC}$ uses the CLS token embedding similar to the IC network, whereas $C_{\Omega,NER}$ applies a single long short-term memory (LSTM) layer to extract longitudinal information from the BERT hidden states at each token.

We also train a DG approach combining DMG and OT (referred to as *Combined*). We retain the critic networks from OT, and introduce masks at the input of critic networks in addition to masks at the inputs of IC and NER networks. The overall loss function to be optimized is a sum of classification losses, critic loss and the overlap penalty loss. We explore whether we can obtain any gains in task performance due to the complementary nature of these approaches.

We use AdamW (Loshchilov and Hutter, 2018) optimizer (initial LR: 5e-5, decay rate: 0.96, (β_1, β_2) = (0.9, 0.999), ϵ = 1e-8) to minimize the respective loss objectives for each approach. We train the models for 10 epochs for SNIPSesv and 5 epochs for TOPesv. To improve training stability, we accumulate gradients from two mini-batches before back-propagation. We follow Chattopadhyay et al. (2020) and Zhou et al. (2020a) to fix approach-specific learning parameters: we set $\lambda_{DMG} = 0.1$ (Eq. 3) and set the critic coefficient as a function of the training progress p, $\lambda_{OT} = \frac{2}{1+e^{-\delta p}} - 1$ where δ = 10. We apply dropout with the rate of 0.1 at all layers in F_Ψ and T_Θ. Following (Chen et al., 2019), we use two metrics to evaluate IC-NER performance: (1) slot-filling F_1 (*Slot* F_1), which is the weighted average of F1 scores across slot labels and (2) semantic accuracy rate (*Sem Acc*), which computes the exact match accuracy of ordered slot labels prefixed with the intent label.

5 Results

5.1 Performance on Seen and Unseen Segments

We report IC-NER performance on the test sets from all four segments in Table 3. For each segment and method, we report mean *Slot* F_1 and *Sem Acc* over 5 trials with different random seeds. We observe that for both datasets, performance on the head segment differs substantially between approaches. Note that in SNIPesv, different approaches produce the same evaluation fig-

ures, which we attribute to the limited number of unique utterances in the head segment (Table 2), even though it contains roughly the same utterance count as the tail. While DG approaches do not provide a boost in performance over baselines for the head segment, this is not necessarily a cause for concern. We believe that in a real-world scenario with digital assistants, very frequent requests can be easily recognized using non-statistical models such as rules and deterministic finite-state-transducers (Mohri, 1997).

Among the three segments, improvements with DG approaches (DMG++, OT & Combined) are more visible in tail: the best DG approach returns 7.02% and 1.27% relative improvement in semantic accuracy and slot F_1 on SNIPSesv datasets over the best performing baseline. The original test set, which is not modified by our work and represents yet another segment demonstrates minor but consistent improvements in both metrics across SNIPSesv and TOPesv. Further, we observe competitive performance by optimal transport-based approaches (OT and Combined) on the body segment: upto 2.25% relative improvement with the best performing baseline on SNIPSesv and identical performance on TOPesv.

We observe that improvements in TOPesv are lesser than SNIPSesv, specifically for Tail and Body segments. We believe that there exists a clearer variation between segments in case of SNIPSesv due to a wider range of topics spanned by the utterances (music, books, events, weather) whereas TOPesv intents are generally confined to navigation. Hence, DG approaches are more likely to exhibit gains over baselines in SNIPSesv vs TOPesv.

5.2 Analysis of DG performance gains

5.2.1 Segment Classification Model

Since OT attempts to learn segment-invariant representations, we validate this paradigm by building a segment classifier on the representations from the trained feature encoder. We extract CLS token embeddings for the above approaches and train a multi-class linear regression model using the segment as class information. We downsample the body segment by a factor of 8 to ensure a uniform class distribution. The per-segment approach trains a different F_Ψ for each segment, hence we compute the mean embedding from all three models. We report segment accuracy (%) in Table 5.

We observe that the approaches which learn segment-specific network components such as per-segment (F_Ψ) and multi-head (T_Θ) yield relatively high classification accuracy, while the aggregate model which learns a single network across segments returns the lowest performance among baselines. Optimal transport performs the worst, suggesting that it learns the least segment-related information. However, the difference with the majority baseline ($\approx 33\%$) suggests that segment-invariant representations may not be completely achieved on the test set, also observed in Galstyan et al. (2022).

5.2.2 Random-valued Mask Analysis

In order to analyze the segment-specific masks learned by DMG++ approach, we compare the learned masks using three metrics: (i) M1: Mean pairwise cosine distance between $\tilde{\mathbf{m}}^d$, (ii) M2: Mean pairwise cosine distance between m^d, and (iii) M3: Mean fraction of "off" (0) dimensions in m^d. Since m^d is sampled from $\tilde{\mathbf{m}}^d$ (Eq. 2), we compute M2 and M3 over 5 trials and report their mean and standard deviation. Note that we only analyze $\tilde{\mathbf{m}}^d_{IC}$ since $\tilde{\mathbf{m}}^d_{NER}$ is dependent on token embeddings.

From Table 6, we notice that $\tilde{\mathbf{m}}^d$ are clearly different between segments in both SNIPSesv and TOPesv. These differences extend to the sampled versions (which are used in forward-pass) are illustrated in M2 and M3, a result of the overlap penalty. Further, masks from all segments are "on" ($= 1$) for $\approx 59\%$ and $\approx 53\%$ dimensions for SNIPSesv and TOPesv respectively. To ascertain if segment-specific information is learned by masks, we conduct a sanity check experiment where we replace the masks with a random parameter that encourages similar fraction of "on" dimensions to the learned masks.

Surprisingly, we notice that random masks return on-par performance on all metrics and segments with the learned masks on both SNIPSesv and TOPesv corpora (Table 4). This result clearly indicates that the masks do not provide segment-specific information and the exact set of "on"/"off" dimensions which are controlled by the learned masks are not critical for performance on unseen segments. To further ascertain this finding, we repeated the random masks experiment on PACS corpora (Li et al., 2017) from computer vision, following (Chattopadhyay et al., 2020), with similar results (see Appendix).

Instead of learning segment-specific information

Table 4: Comparing IC-NER performance between learnt masks (**DMG**) and random masks (**DMG-Random**; repeated over 10 trials) on SNIPSesv and TOPesv. For brevity, only semantic accuracy (Sem) and slot filling F1 (Slot F1) are presented

			Head		**Body**		**Tail**		**Original**	
Dataset	**Approach**		**Sem**	**SlotF1**	**Sem**	**SlotF1**	**Sem**	**SlotF1**	**Sem**	**SlotF1**
SNIPSesv	DMG++	-	87.84	96.73	88.33	95.59	88.41	93.87	87.00	94.74
	DMG-	μ	78.65	95.55	88.26	95.58	87.97	93.66	86.74	94.67
	Random	σ	2.55	0.33	0.18	0.05	0.67	0.29	0.18	0.08
TOPesv	DMG++	-	88.93	97.06	90.18	95.94	86.81	93.91	89.03	94.63
	DMG-	μ	88.80	96.96	90.08	95.85	86.83	93.86	88.91	94.55
	Random	σ	0.45	0.26	0.18	0.06	0.26	0.23	0.11	0.09

Table 5: Segment classification accuracy (%) for baselines and optimal transport. Majority baseline: $\approx 33\%$

	Per	Agg	Mul	OT
SNIPSesv	91.03	86.03	90.13	69.36
TOPesv	79.22	72.33	76.78	65.56

Table 6: Comparing learnt ($\tilde{\mathbf{m}}^d$) and sampled mask (m^d) parameters across segments

Metric	SNIPSesv	TOPesv
M1	0.41	0.95
M2	0.41 ± 0.03	0.53 ± 0.01
M3	40.76 ± 1.57	52.70 ± 1.31

as suggested by Chattopadhyay et al. (2020), we believe that the improvements yielded by DMG approach can be attributed to learning generalizable parameters using masks. Masks are encouraged to be robust by the training process, since m^d are stochastically determined at each mini-batch even for samples from the same segment. Further, our experiments with random masks resemble the training process in that a different set of masks are sampled, except that gradients are not back-propagated. Finally, we note that sampled masks operate similar to a segment-specific dropout (Srivastava et al., 2014) strategy. Hence, generalization improvements in deep learning which have been observed by dropout are likely to be enhanced with segment-specific mask parameters.

6 Limitations

Obtaining search volumes using the Google Adwords API cannot disambiguate between different context-based semantic interpretations of the same word, especially when there are no additional tokens to provide context. For instance, search volumes for *apple* will combine volumes related to the corporation and the fruit, while *apple phone* and *apple juice* will return only the relevant search volumes. Further, this work did not address availability concerns for tail utterances/entities which may be more expensive or labor intensive to collect and annotate.

7 Conclusions

We presented a methodology to estimate utterance frequency information in public datasets for IC-NER task. We create two new corpora: SNIPSesv and TOPesv which use the frequency information to segment the original corpora into head, body and tail segments. We adapt two DG approaches for IC-NER and compute performance on each segment as well as the original test set, which represents an unseen segment. Our experiments show improvement in tail entity recognition by each DG approach as well as their combination. Our follow-up analyses validate the segment-invariant representation learning by OT and suggest that DMG provides enhanced generalization using segment-specific masks. To assist future research in this direction, we will release the SNIPSesv and TOPesv datasets used in this work upon publication.

References

Raviteja Anantha, Srinivas Chappidi, and William Dawoodi. 2021. Learning to rank intents in voice assistants. In *Conversational Dialogue Systems for the Next Decade*, pages 87–101. Springer.

Rodrigo Berriel, Stephane Lathuillere, Moin Nabi, Tassilo Klein, Thiago Oliveira-Santos, Nicu Sebe, and Elisa Ricci. 2019. Budget-aware adapters for multi-domain learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 382–391.

Gilles Blanchard, Aniket Anand Deshmukh, Urun Dogan, Gyemin Lee, and Clayton Scott. 2021. Domain generalization by marginal transfer learning. *Journal of Machine Learning Research*, 22:1–55.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. 2011. Generalizing from several related classifica-

tion tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*.

Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. 2020. Learning to balance specificity and invariance for in and out of domain generalization. In *European Conference on Computer Vision*, pages 301–318. Springer.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. BERT for joint intent classification and slot filling. *CoRR*, abs/1902.10909.

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. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR*, abs/1805.10190.

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*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Tigran Galstyan, Hrayr Harutyunyan, Hrant Khachatrian, Greg Ver Steeg, and Aram Galstyan. 2022. Failure modes of domain generalization algorithms. *CoRR*, abs/2111.13733.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. 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. Association for Computational Linguistics.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. 2020. Decoupling representation and classifier for long-tailed recognition. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Leonid Vasilevich Kantorovich and SG Rubinshtein. 1958. On a space of totally additive functions. *Vestnik of the St. Petersburg University: Mathematics*, 13(7):52–59.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. 2017. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Arun Mallya, Dillon Davis, and Svetlana Lazebnik. 2018. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 67–82.

Gabriel Marzinotto, Géraldine Damnati, Frédéric Béchet, and Benoît Favre. 2019. Robust semantic parsing with adversarial learning for domain generalization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 166–173. Association for Computational Linguistics.

Mehryar Mohri. 1997. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. 2013. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR.

Wanli Ouyang, Xiaogang Wang, Cong Zhang, and Xiaokang Yang. 2016. Factors in finetuning deep model for object detection with long-tail distribution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 864–873.

Md Atiqur Rahman and Yang Wang. 2016. Optimizing intersection-over-union in deep neural networks for image segmentation. In *International symposium on visual computing*, pages 234–244. Springer.

Vikas Raunak, Siddharth Dalmia, Vivek Gupta, and Florian Metze. 2020. On long-tailed phenomena in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3088–3095.

Sachin Ravi and Hugo Larochelle. 2017. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. 2018. Generalizing across domains via cross-gradient training. *CoRR*, abs/1804.10745.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Chengwei Su, Rahul Gupta, Shankar Ananthakrishnan, and Spyros Matsoukas. 2018. A re-ranker scheme for integrating large scale nlu models. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 670–676.

Bailin Wang, Mirella Lapata, and Ivan Titov. 2021. Meta-learning for domain generalization in semantic parsing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–379.

Lin Xiao, Xiangliang Zhang, Liping Jing, Chi Huang, and Mingyang Song. 2021. Does head label help for long-tailed multi-label text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14103–14111.

Fan Zhou, Zhuqing Jiang, Changjian Shui, Boyu Wang, and Brahim Chaib-draa. 2020a. Domain generalization with optimal transport and metric learning.

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. 2020b. Deep domain-adversarial image generation for domain generalisation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13025–13032.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. 2021. Domain adaptive ensemble learning. *IEEE Transactions on Image Processing*, 30:8008–8018.

A Determining Maximum Utterance Sampling Probability

We collected a real-world dataset of user-queries directed to our voice-controlled agent to determine the maximum utterance sampling probability p_{max}. We uniformly sample from all queries within a 10-day duration to preserve the frequency distribution. However, we retain only utterances which were identified as belonging to services similar to intents in SNIPS and TOP corpora: entertainment (music, books, video), weather, bookings and local search. This results in a total of 15M utterances. We compute repetition counts for each unique utterance and compute p_{max} using the utterance with maximum repetition count following Eq. 1. This results in p_{max}=0.00245. We apply this estimated value for P_{max} on SNIPSesv and TOPesv.

B Random-valued Masks for PACS

PACS corproa (Li et al., 2017) is a commonly used DG benchmark from computer vision and contains images from four different styles: sketch, cartoon, photo and art painting. Similar to previous evaluations (Li et al., 2017; Chattopadhyay et al., 2020; Zhou et al., 2020a), we compute the leave-one-domain-out accuracy, where one domain is treated as target and remaining three domains are treated as source. We build a DMG model following the same architecture as (Chattopadhyay et al., 2020) and repeat our evaluations by replacing the learned masks with random valued parameters. We observe identical performance with random masks, similar to SNIPSesv and TOPesv.

Table 7: Leave-one-domain-out accuracy (%) on PACS. DMG (rep) represents results reported in Chattopadhyay et al. (2020), DMG (ours) reports results from our implementation, and DMG (rand) uses random valued masks.

Approach		Sketch	Cartoon	Photo	Art
DMG (rep)		71.42	69.88	87.31	64.65
DMG (ours)		67.98	67.83	84.25	63.48
DMG (rand)	μ	67.24	67.71	83.75	63.19
	σ	0.32	0.06	0.13	0.24