Comprehensive Bench-marking of Entropy and Margin Based Scoring Metrics for Data Selection

Anusha Sabbineni[1] **Nikhil Anand**[1] **Maria Minakova**[2]
[1]Alexa AI (Amazon) [2]Work done at Amazon
{sabanu,nkhlanan}@amazon.com
{maria.s.minakova}@gmail.com

Abstract

While data selection methods have been studied extensively in active learning, data pruning, and data augmentation settings, there is little evidence for the efficacy of these methods in industry scale settings, particularly in low-resource languages. Our work presents ways of assessing prospective training examples in those settings for their "usefulness" or "difficulty". We also demonstrate how these measures can be used in selecting important examples for training supervised machine learning models. We primarily experiment with entropy and Error L2-Norm (EL2N) scores. We use these metrics to curate high quality datasets from a large pool of *Weak Signal Labeled* data, which assigns no-defect high confidence hypotheses during inference as ground truth labels. We then conduct training data augmentation experiments using these de-identified datasets and demonstrate that score-based selection can result in a 2% decrease in semantic error rate and 4%-7% decrease in domain classification error rate when compared to the baseline technique of random selection.

1 Introduction

The immense progress in deep learning over the past decade has been, in part, driven by the increasing scale of training data (Hoffmann et al., 2022), model architectures like transformers Vaswani et al. (2017), and compute used to train the models. However, the science of surfacing *which* examples to include in training data remains a persistent and applicable question. The recently published Platypus family of models (Lee et al., 2023b) outperformed several SOTA open Large Language Models(LLMs) while being trained on a single GPU for only five hours. The reported success at such a low cost appears primarily due to the quality of their smaller dataset which was curated from large pools of open datasets, which reiterates the significance of curating high quality datasets for model training.

Many studies have been conducted on data selection from large pools of data, but there are challenges when it comes to implementing them in large scale systems. One challenge is that they are studied in an offline, one-time selection setting from a static data pool while most real world systems need to implement data selection on a continual basis with potential data drift. To make data selection practical and scalable, it should be based on scores that are easy to compute and interpret, stay relevant with changes to data distribution and model architectures, and can be integrated into existing data collection pipelines easily. A second challenge is that some data selection techniques are too compute intensive (Ash et al., 2019) to be easily implemented and integrated. Third, a few selection methods like "Selection Via Proxy" (Coleman et al., 2019) require training and maintaining proxy models for data selection, which adds to the resources overhead. In the context of langauge models (LMs), recent studies for data selection experimented with changes to pre-training, such as Task Adaptive Pre-Training (TAPT) (Margatina et al., 2021). These approaches are not architecture agnostic and

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

the complimentary models need to be retrained with changes to underlying training data, prohibiting continuous data selection without interruptions.

The above mentioned challenges become more pronounced when dealing with unstructured data in low resource languages. Most data selection methods are developed where training data is in English and experiments on non-English languages are typically conducted after the methods are optimized on English data. While this is a gap in research, it also raises the question, "Do these selection methods work for training models on low-resource languages? If so, how well?" There have been studies on data selection methods in multilingual settings for Neural Machine Translation (van der Wees et al., 2017). Hedderich et al. (2021) surveyed methods that enable learning when training data is sparse which includes data augmentation. However, data selection strategies for unstructured data in low-resource languages for supervised machine learning models is relatively a less explored area of research.

All the above challenges and issues emphasize the need to implement efficient and scalable data selection methods for low resource languages. We conduct experiments on Portuguese language. Our work aims to test two metrics – entropy and EL2N – in large scale conversational systems. The implementation of these metrics are agnostic to changes to data distribution and model architecture. They are interpretable, easy to compute, scalable, and can be integrated into existing pipelines that involve routine data selection. Our work presents a comprehensive benchmarking of improvements from score based data selection methods, dives deep into how those affect training data, and the overlap of the data selected using different methods for supervised machine learning models.

We conduct our experiments on a BERT based model (Devlin et al., 2018) that performs domain, intent and slots recognition. Components that are targeted for improvements are detailed in sec. A.

2 Scoring metrics

Entropy: Natural Language Understanding (NLU) is a key component in a conversational system. Domain Classifier (DC) within NLU predicts the domain a user request needs to be routed to. A DC can be expressed as a function parametrized by a set of weights θ : $f_\theta^j(x)$ is the softmax output for j belonging to one of N classes. For some input x, we compute entropy (Shannon, 1948) of DC outputs for a single example as shown in the eq. 1.

$$H(x) = -\sum_{j=0}^{N-1} f_\theta^j(x) \log_2 f_\theta^j(x) \qquad (1)$$

$$\texttt{EL2N}(x) = \|\boldsymbol{f}_\theta(x) - \boldsymbol{y}_i\|_2 \qquad (2)$$

EL2N is a margin-based metric that estimates gradient norms as described in Paul et al. (2021) . It is computed as in eq. 2, where $\boldsymbol{f}(x)$ indicates the softmax of the model outputs and $\boldsymbol{y}_i$ indicates the one-hot encodings of the label for the ith example. Typically, this metric is averaged over $\mathcal{O}(1)$ number of "replicates" (model initializations) to obtain a reliable signal of example difficulty. We compute EL2N scores at fine-tuning of DC, averaged over five replicates for each example.

Predictions from DC, along with intent and token classification, feed as inputs to downstream tasks in a NLU system. Any improvements to DC could translate to improvement in the entire system. So, we anchored our experiments on data selection with EL2N and entropy based on DC outputs.

NLU Model Confidence Score: In the conversational system under experimentation, NLU Model Confidence Score, referred to as "NLU Score" from here on, is a measure of the system's confidence in the suggested hypothesis i.e, predicted classes (domain and intent) and labels recognition (for individual tokens). NLU Score is a calibrated metric with a range (0,1]. DC scores i.e, softmax outputs from the classifier, are one set of inputs to NLU score. Other inputs include, but are not limited to, intent classifier and token recognition scores. In general, a hypothesis with the correct domain should have a high NLU Score and vice-versa. NLU score is calculated per domain. Unlike softmax outputs from a classifier, NLU scores from the system across domains don't add to 1.

3 Datasets

Existing training data is the de-identified data used to train an in-house model used for experiments.

New dataset is an inhouse Weak-Signal Labeled (WSL) dataset sourced using the procedure described in Schroedl et al. (2022), where data is constructed from *weak supervision* from the user to obtain NLU labels. For example, if an unsuccessful action from the voice assistant is followed by a user rephrasing their request, which then results in an uninterrupted response from the device, that utterance is pseudolabeled using the top NLU hypothesis. We begin with a de-identified WSL dataset with NLU Score range [0.3,0.85] collected over a period of time, referred to as "new dataset" from here on. The selected range aims to eliminate examples that are already well learnt by the model (NLU score > 0.85) and noisy/ambiguous examples (NLU Score < 0.3). New dataset comes from de-ientified live traffic with its size quite larger when compared to existing training data, and is heavily skewed towards popular user interactions i.e., less diverse. We experimented with entropy and EL2N score based data selection strategies to curate smaller datasets from the new dataset with most useful/difficult examples. These curated datasets are added to the existing training data to build different candidate models as listed in sec. 4.1. Figure 1 shows data distribution of top seven domains, accounting to 90% user traffic, in existing training data (left) and new dataset (right). Figure 2 shows the distribution of datasets curated using score based selection methods detailed in sec. 4.1.

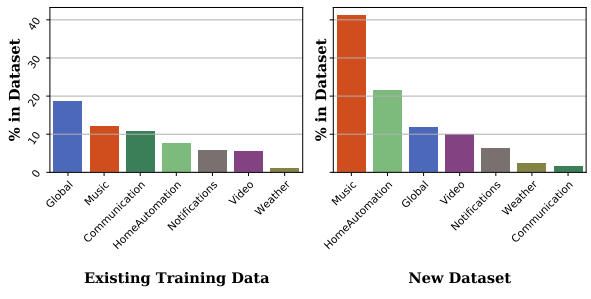

Figure 1: Data distribution of top seven domains in the existing training data (left) and new dataset (right); Both plots share y-axis.

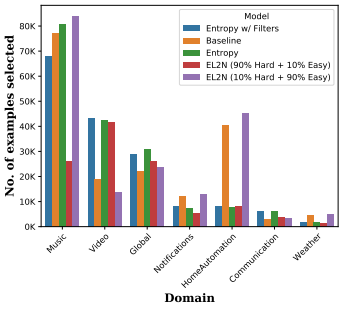

Figure 2: Data distribution of top seven domains in the datasets curated using different data selection methods

Dataset curation based on Entropy: For each sample in the new dataset we generate DC scores for possible domains using the eq. 1, and calculate entropy of the generated DC scores. We then rank all the examples based on their entropy scores and select top K examples to create a smaller dataset. Sec. C.1 has exact details of how the examples are selected based on entropy.

Dataset curation based on EL2N: For each training example in the dataset, we generate EL2N scores using eq. 2 averaged over five replicates. The scores are generated through a domain classification task. After the data was ranked by EL2N difficulty, we use a threshold score of ≤ 0.15 to denote "easy" examples and a threshold of ≥ 0.6 to denote "hard" examples. We then randomly sampled from these subsets for a given target dataset size to obtain different mixtures of easy and hard examples, with splits described in sec. 4.1.

Specialized testsets: We use specialized test sets to evaluate model performance for the use cases of interest. Our objective is to improve generalizability of the model, and so we evaluate our models on specially curated test sets that have the same distribution as the tail 40% of user traffic and are de-identified.

4 Experiments

4.1 Models

We take one of the in-house models that performs NLU tasks as a baseline to experiment for further accuracy improvements. The in-house model is retrained with different additional datasets, all of the same size, resulting in different candidate models as listed below.

Baseline model has existing training data augmented with randomly selected data from new dataset.

EL2N (10% Hard + 90% Easy) candidate has training data augmented with data selected based on EL2N scores from the new dataset with a composition of 10% hard examples and 90% easy examples. Section 3 has details on how easy and hard examples are selected.

EL2N (90% Hard + 10% Easy) candidate has training data augmented with data selected based on EL2N scores from the new dataset with a composition of 90% hard examples and 10% easy examples. Section 3 has details on how easy and hard examples are selected.

Entropy candidate has training data augmented with data selected based on entropy scores of domain classifier outputs as detailed in sec. 3. Examples with higher entropy scores are added to training data. Sec. C.1 has exact details of why and how the examples were selected based on entropy.

Entropy with Filters candidate has additional data processing on the dataset curated based on entropy scores. First, we limit the repetition of an example in the curated dataset to P to enhance the diversity. We observed that with no limit in place, approx. 4,000 examples selected for a domain came from two unique examples. Second, we ensure that each domain has a minimum representation of $R\%$ in the dataset. Our values for P and R are 20 and 0.5. Further discussion can be found at sec. C.1.

4.2 Evaluation Metrics

We measure candidates' performance on specialized test sets introduced in sec. 3 in terms of component-wise (domain, intent and slots) error rates. In terms of component-wise error rates, we measured domain classification performance using the recall-based classification error rate DCER. To evaluate slot-filling performance, we measured semantic error rate (SEMER):

$$\texttt{SEMER} \equiv \frac{\texttt{\# Intent errors} + \texttt{\# Slot errors}}{\texttt{\# Test data} + \texttt{\# Slots}}.$$

We measured F-SEMER, the harmonic mean of SEMER using predicted labels as the reference and SEMER computed on ground-truth labels as the reference; this score balances precision/recall equally. We also report the interpretation error rate IRER, which reflects the rate of any kind of error (slots, intents, domain).

5 Results

Table 1 presents evaluation results for different candidates listed in the section 4.1. All the values are relative to a baseline model trained on randomly sampled data from the new dataset. We see improvement across all metrics for all candidates except *EL2N (90% Hard + 10% Easy)*. Improvements to SEMER, F-SEMER and IRER are in the order of 2% with respect to the baseline. Improvements to DCER are in the range of 3%-7%. *Entropy w/ Filters* candidate and *EL2N (10% Hard + 90% Easy)* have the most promising results.

Table 1: Evaluation results relative to a baseline with random data selection

Model	Δ SEMER% $\downarrow$	Δ F-SEMER% $\downarrow$	Δ DCER% $\downarrow$	Δ IRER $\downarrow$
Entropy	-2.35	**-2.29**	-6.11	-1.46
Entropy w/ Filters	**-2.37**	-2.24	**-7.20**	-0.55
EL2N (90% Hard + 10% Easy)	-0.08	-0.16	3.09	-0.06
EL2N (10% Hard + 90% Easy)	-2.14	-2.16	-4.12	**-2.10**

Table 3 presents more nuanced results when we look at domain level metrics. One can notice that Video, Notifications, Weather and Communications domains are better served by entropy based data selection while Music and Home Automation domains are better served by EL2N score based data selection. We recommend avoiding "one-size-fits-all" approach and encourage identifying which technique performed the best for each domain. Once the initial experiments are done, domains can be mapped to different data selection pipelines. Sec. D.1 and sec. E have further discussion on metrics and data selected.

6 Conclusion

Entropy based data selection improved metrics (DCER) better than EL2N on anchor task (DC), which was the source for the score. EL2N, however, delivered better improvements to overall recognition as measured by IRER. We hypothesize that EL2N, averaged over multiple runs, captures example importance towards overall accuracy, while entropy is better suited to improve a specific task. With growing developments in LLMs, we plan to continue our work on data selection strategies for LLMs, and using LLMs for data selection. Our areas of research would be fine-tuning LLMs, in-context learning (Brown et al., 2020), avoiding model collapse (Shumailov et al., 2023), and experimenting with a broader set of metrics (Marion et al. (2023), (Lee et al., 2023a)) for data selection.

References

J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners, 2020.

C. Coleman, C. Yeh, S. Mussmann, B. Mirzasoleiman, P. Bailis, P. Liang, J. Leskovec, and M. Zaharia. Selection via proxy: Efficient data selection for deep learning. *arXiv preprint arXiv:1906.11829*, 2019.

J. Devlin, M.-W. Chang, K. Lee, and K. N. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018. URL `https://arxiv.org/abs/1810.04805`.

M. A. Hedderich, L. Lange, H. Adel, J. Strötgen, and D. Klakow. A survey on recent approaches for natural language processing in low-resource scenarios, 2021.

J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae, O. Vinyals, and L. Sifre. Training compute-optimal large language models, 2022.

N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

A. Lee, B. Miranda, and S. Koyejo. Beyond scale: the diversity coefficient as a data quality metric demonstrates llms are pre-trained on formally diverse data, 2023a.

A. N. Lee, C. J. Hunter, and N. Ruiz. Platypus: Quick, cheap, and powerful refinement of llms. *arXiv preprint arXiv:2308.07317*, 2023b.

K. Margatina, L. Barrault, and N. Aletras. On the importance of effectively adapting pretrained language models for active learning. *arXiv preprint arXiv:2104.08320*, 2021.

M. Marion, A. Üstün, L. Pozzobon, A. Wang, M. Fadaee, and S. Hooker. When less is more: Investigating data pruning for pretraining llms at scale, 2023.

M. Paul, S. Ganguli, and G. K. Dziugaite. Deep learning on a data diet: Finding important examples early in training. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 20596–20607. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/ac56f8fe9eea3e4a365f29f0f1957c55-Paper.pdf`.

S. Schroedl, M. Kumar, K. Hajebi, M. Ziyadi, S. Venkatapathy, A. Ramakrishna, R. Gupta, and P. Natarajan. Improving large-scale conversational assistants using model interpretation based training sample selection. 2022.

C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 1948. URL http://plan9.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf.

I. Shumailov, Z. Shumaylov, Y. Zhao, Y. Gal, N. Papernot, and R. Anderson. The curse of recursion: Training on generated data makes models forget, 2023.

M. van der Wees, A. Bisazza, and C. Monz. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1147. URL https://aclanthology.org/D17-1147.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Y. Yu, L. Kong, J. Zhang, R. Zhang, and C. Zhang. Actune: Uncertainty-based active self-training for active fine-tuning of pretrained language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1422–1436, 2022.

A Environment:

We conducted our experiments on a conversational system. A typical conversational system has speech and Natural Language Understanding (NLU) components along with many others such as business logic based hypothesis re-routing, invoking third party applications, and more. Our experiments target improvements to NLU components. Within NLU, we need to recognize multiple things correctly to deliver desired experience to end users. A few of them are *domain, intent, and slots*. In the example, "Play Taylor Swift", *domain* is "Music", *intent* is "Play Music" and *slot_name* is "Artist Name" with a *slot_value* of "Taylor Swift". In our results, we show how our techniques improved metrics on each of these recognition tasks. Our experiments are anchored on domain classifier outputs.

B Exploratory Data Analysis

B.1 Correlation study of NLU scores and Entropy scores

We did a correlation study between NLU scores and corresponding entropy of domain classifier scores on a pool of 5 million examples. Results are shown in Table 2.

All the three correlation coefficients show a consistent negative correlation between model confidence scores and entropy scores which implies that by reducing entropy of DC scores, we could improve NLU models' confidence on the correct interpretations and serve the end users with the most relevant responses.

Table 2: Correlation coefficients between entropy of domain classifier scores and NLU scores

Correlation Coefficient	Value
Pearson Correlation	-0.3132
Spearman's Rank Correlation	-0.0732
Kendall tau-a	-0.104

B.2 Distribution of Entropy and EL2N on the large new dataset

Both Entropy and EL2N metrics have long tail distributions on the new dataset (Figure 3). Most of the examples have low scores implying relative high certainty in prediction (low entropy) or relative ease of getting correct prediction (low EL2N). We can observe that Entropy and EL2N scores distributions are left skewed in the Figure 3 while NLU score distribution in the Figure 4 is right skewed re-iterating the they trend in the opposite directions. Hence, reducing entropy or EL2N

scores on a dataset could improve NLU score and eventually the system. The long tail presents a real opportunity to select and add challenging examples to training data. Models benefit from learning from these challenging examples during training which could result in improved accuracy.

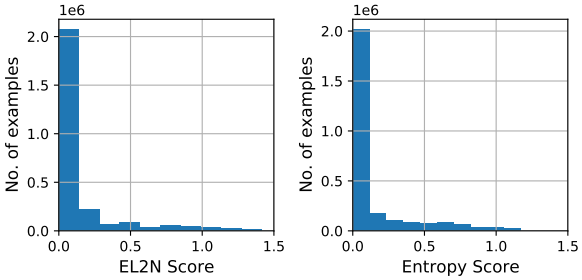

Figure 3: EL2N and Entropy scores distribution on the new dataset.

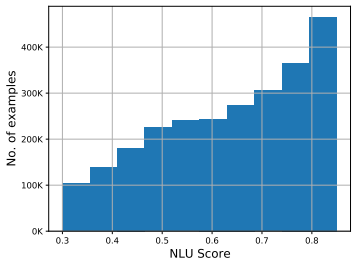

Figure 4: NLU Score distribution in the new dataset

Range of Entropy for the full dataset after removing outliers (z-score > 3) is [0.0, 1.1731] with a mean and standard deviation of 0.137 and 0.252 respectively. Range of EL2N score for the full dataset after removing outliers (z-score > 3) is [0.0, 1.414] with a mean and standard deviation of 0.128341 and 0.275945 respectively.

C Scoring metrics

Entropy H of a random variable X is the level of uncertainty inherent in the variable's possible outcomes. In case of a classification problem, $p_j(x_i)$ is the probability of an example x_i belonging to the class *j*. The entropy of a single example x_i that could belong to *n* possible classes is calculated as

$$H(X = x_i) = -\sum_{j=1}^{n} p_j(x_i) \log_2 p_j(x_i) \tag{3}$$

C.1 Motivation to use Entropy for data selection

In Active Learning (AL), entropy is used as an acquisition function to find the most *informative* examples to label from a large pool of unlabelled data. Entropy captures the uncertainty associated with the predicted labels. The higher the entropy of predictions, the higher is the possibility that the model is *more challenged or confused* when presented with such samples. Samples with higher entropy are selected for annotations. Those annotated examples are then added to training data to improve the models. In AL, Entropy based selection is proved to work better than many other sophisticated acquisition functions like BADGE (Ash et al. (2019)) and BALD (Houlsby et al. (2011)) when tested on a variety of tasks. Recent methods like AcTune (Yu et al. (2022)) show that entropy when employed with other scientific techniques delivers superior results. Drawing inspiration from entropy based data selection for labeling in AL, we implemented entropy as a metric to select data from a large pool of *Weak-Signal Labeled(WSL)* data. More information on WSL data is presented in section 3 . One more reason to choose entropy over other methods is the ease of implementation and cost to compute.

Selection based on entropy: For any new dataset *D*, we compute entropy of domain classification softmax outputs, referred to as "entropy". We rank examples in *D* in decreasing order of their entropy and select top *K* examples based on a given cut off criteria. Intuition behind this is that we train the models with data that the models are less certain about but would benefit from learning from those during training. This should result in improved accuracy of the model predictions. Our cut off criteria was to limit the number of additional training examples that can be added to the existing training data. We experimented with newly created datasets to be in the order of 2% or 5% of existing training data. Datasets of size 5% gave the best results. Other selection criteria could be cut off based on absolute entropy score, top X% of examples from the larger dataset, etc.

For **Entropy w/ Filters** candidate, we chose values of 20 and 0.5% for an example repetition cap and minimum representation of a domain (class). We arrived at these values after multiple experiments. There values can and should be experimented for different modeling tasks and datasets.

D Results

D.1 Metrics by Domain

Table 3 presents evaluation results for the top seven domains (classes) that account to 90% user traffic. Results show that different domains benefit from different data selection strategies. We recommend avoiding "one-size-fits-all" approach and encourage identifying which technique performed the best for each domain. Once the initial experiments are done, domains can be mapped to different data selection pipelines. Looking at the Table 3, one can notice that Video, Notifications, Weather and Communications domains are better served by entropy based data selection while Music and Home Automation domains are better served by EL2N score based data selection. For domain level metrics, we reported balanced metrics for DCER and IRER i.e, F-DCER and F-IRER along with SEMER that was introduced in sec. 4.2

Table 3: Evaluation results by domain relative to a baseline with random data selection

Domain	Model	Δ SEMER %	Δ F-DCER%	Δ F-IRER
Music	Entropy	-5.15	-5.38	-1.92
	Entropy w/ Filters	-3.34	-5.83	-0.41
	EL2N (90% Hard + 10% Easy)	-1.80	7.17	-1.11
	EL2N (10% Hard + 90% Easy)	**-4.03**	**-5.83**	**-2.85**
Video	Entropy	1.26	-10.03	-3.99
	Entropy w/ Filters	**-1.73**	**-11.34**	**-3.99**
	EL2N (90% Hard + 10% Easy)	1.73	1.97	1.56
	EL2N (10% Hard + 90% Easy)	3.06	-4.98	-1.82
Home Automation	Entropy	-0.93	-4.07	1.43
	Entropy w/ Filters	-2.51	-6.50	2.90
	EL2N (90% Hard + 10% Easy)	1.02	0.81	1.52
	EL2N (10% Hard + 90% Easy)	**-3.62**	**-5.28**	**-2.43**
Global	Entropy	-4.82	-0.12	2.14
	Entropy w/ Filters	1.08	-1.09	0.78
	EL2N (90% Hard + 10% Easy)	1.58	7.13	3.18
	EL2N (10% Hard + 90% Easy)	2.81	-2.06	-0.71
Notifications	Entropy	-4.18	1.45	2.79
	Entropy w/ Filters	**-6.07**	**-0.72**	**2.79**
	EL2N (90% Hard + 10% Easy)	2.72	25.36	2.15
	EL2N (10% Hard + 90% Easy)	-5.23	-0.72	0.75
Weather	Entropy	**-3.50**	**-10.46**	**0.69**
	Entropy w/ Filters	-3.15	-4.58	5.02
	EL2N (90% Hard + 10% Easy)	3.85	1.96	7.49
	EL2N (10% Hard + 90% Easy)	2.45	-3.92	3.15
Communications	Entropy	-0.50	2.27	1.07
	Entropy w/ Filters	**-0.83**	**-1.55**	**-0.09**
	EL2N (90% Hard + 10% Easy)	1.87	1.09	1.86
	EL2N (10% Hard + 90% Easy)	0.96	-1.18	0.42

E Analysis

E.1 Data Distribution of Selected Training Examples by Each Method

Figure 2 shows the data distribution of top seven domains, accounting to 90% user traffic, in the datasets curated based on different data selection methods. Baseline method randomly selects examples from the new dataset and is representative of user traffic. Datasets curated for *Entropy* and *Entropy w/ Filters* have similar distribution across domains. This is corroborated by entropy datasets' overlap of 89.9% presented in Table 4. These two methods tend to surface diverse examples on which the models are less certain about the predictions. As a result, their distribution could vary from baseline's. We observe that entropy based methods picked less no. of examples for the domain *HomeAutomation* relative to baseline as the user interactions are relatively more standard i.e., less diverse. An example interaction for *HomeAutomation* is "Turn of the lights". They picked more no. of examples for *Video, Global* and *Communication* given the diverse nature of user interactions. Example interactions for *Video* are "Play my favorite movie" and "I want to watch Harry Porter". Datasets from EL2N based selection don't have similar distributions unlike entropy based selection. This is corroborated by EL2N datasets' overlap of 19.7% presented in Table 4. It is interesting to note that EL2N (10% Hard + 90% Easy) selected more examples in six out of seven domains under review relative to baseline while EL2N (90% Hard +10% Easy) has shown no clear pattern.

E.2 Overlap of data between Entropy and EL2N score based selections

Table 4 presents the overlap of data between datasets curated by different methods. Entropy based datasets had the highest overlap of 89.94%. This is not surprising given they both select examples based on their ranked entropy scores and there is no randomness involved. They only differ for limits of example repetition and minimum domain representation. On the other hand, EL2N based datasets have significantly lower overlap of 19.7% when compared to entropy datasets. This is because these datasets are randomly sampled from different regions of EL2N score distribution. Except for *EL2N (90% Hard + 10% Easy)*, all candidates delivered superior results when compared to the baseline. We hypothesize that this is because the said method deliberately adds "difficult" examples in large volume (90%) which could result in adding more noise than anticipated, and making the model convergence slower within give no. of training epochs.

Table 4: Overlap (percent) of data between datasets curated by different methods

	Baseline	Entropy w/ Filters	Entropy	EL2N (10% Hard + 90% Easy)	EL2N (90% Hard + 10% Easy)
Baseline	1	7.4	7.7	7.6	7.2
Entropy w/ Filters		1	89.94	6.7	36.6
Entropy			1	7.1	33.57
EL2N (10% Hard + 90% Easy)				1	19.7
EL2N (90% Hard + 10% Easy)					1

F Limitations

When opting for entropy based data selection one needs to understand that (i). some level of uncertainty or diversity in responses is inherent in conversational systems and is desired. Response's success is subject to user's preference, location and context. For example, "Resume Harry Porter" could be interpreted as "resume the Harry Porter movie I was watching on my TV earlier" if the user has an active video session in the environment he is interacting with. If not, it could be interpreted as "resume playing an audio book on Harry Porter" if the user is interacting with a screen less device. So, augmenting training data with one particular interpretation where multiple correct responses exist is something practitioners need to watch out for. It is recommended that we augment data with all possible or most relevant interpretations for the same input, (ii) increasing NLU confidence score might not translate to correct prediction in all cases. If we are not watchful, we could be adding noise i.e, examples that are ambiguous or corrupt. This could potentially degrade models' performance. One way to address this problem is to exclude outliers based on entropy score (z-score >= 3). In addition to excluding outliers, we can instate a criteria of minimum repetition in the dataset to avoid selecting such one off unusual user interactions.

When it comes to EL2N based data selection, one needs to experiment with different thresholds and data compositions before deciding on the optimal values. These optimal values could change with model architecture and data drift.