Training Dyamics and Entropy-Aware Mixup for Robust Text Classification

Atharva Kulkarni, Raghav Kapoor, Anubha Kabra
Carnegie Mellon University, Language Technologies Institute
{atharvak, raghavka, anubhak}@cs.cmu.edu

Abstract

Interpolation-based data augmentation techniques such as '*Mixup*' have proven helpful in data-scarce regimes. However, traditional mixup techniques choose samples randomly to generate synthetic data via linear interpolation. In this work, we propose an intelligent mixup technique that pairs data samples based on their difficulty and entropy for generating diverse and robust synthetic examples. To begin with, we identify samples that are *easy-to-learn*, *mislabeled*, and *ambiguous* using the model's training dynamics. Then we proffer a mixup-based data augmentation technique that generates effective synthetic samples by intelligently combining easy and difficult samples based on their entropy in a label-aware fashion. These examples contribute toward improving model robustness and generalization capabilities. Moreover, by discarding the mislabeled examples and training the model with synthetically generated examples of the same size as the original dataset, our method yields about 2% improvement on the challenging iSarcasm dataset. Thus, requiring no extra data, our technique proves to be potent in the low-data setting. We further analyze our results along with ablation experiments. Lastly, we provide qualitative and quantitive analysis to assess our model's performance.

1 Introduction

Modern-day NLP is characterized by large datasets and big benchmarks (Bowman et al., 2015; Rajpurkar et al., 2016; Wang et al., 2018). Though transformer-based language models (Devlin et al., 2019; Liu et al., 2019) have reported exceedingly well results on these benchmarks, they are prone to overfitting on small niche corpora, failing to generalize on unseen data. Data augmentation, especially mixup, has shown promising results in low-data settings. However, efficiently choosing data points for mixup-based interpolation is a difficult task.

Most of the datasets in NLP are either manually curated through human annotation or use weakly-supervised labeling schemes to circumvent the tedious and financially expensive annotation process (Malik and Bhardwaj, 2011). Though increasing the amount of labeled data improves the likelihood of learning diverse data distributions, the data quality plays a significant role in robustness and out-of-distribution generalization. Whether it be automatic labeling or annotation bias, almost all current datasets are interspersed with duplicate, untidy, redundant, mislabelled, corrupted, or ambiguous examples hampering the quality of the dataset. Such label noise can hinder the generalization capabilities of the overparameterized neural networks, owning to their strong proclivity to memorization (Arpit et al., 2017; Zhang et al., 2021). The problem is exacerbated when we have limited training data. Therefore, assessing dataset quality and finding examples that contribute to achieving good performance on in-distribution as well as out-of-distribution examples is vital.

While ambiguity is intrinsic to natural language, it is important to identify data points that contribute to better learning and generalization (Swayamdipta et al., 2020; Karamcheti et al., 2021). Moreover, training on just good samples is not enough; it is essential to include examples from diverse distributions to help generalize better on in-domain as well as out-of-distribution data points (Linzen, 2020). For data augmentation techniques such as mixup, the mannerism in which we pair up examples for interpolation plays a crutial role. Instead of random pairing, using a apriori information such as label and similarity as shown to fare better.

To this end, we propose a novel methodology encapsulating the aforementioned points for robust text classification. To begin with, we propose a training dynamics-based approach that identifies *easy-to-learn*, *ambiguous*, and *hard-to-learn* examples (Swayamdipta et al., 2020) in a dataset. This spawns examples that contribute positively towards faster convergence, examples that help generalize, and mislabelled examples, respectively. To widen the distribution of learn-worthy examples, we suggest a mix-up strategy that uses a combination of easy-to-learn and ambiguous groups to produce high-quality examples. Furthermore, instead of random pairing, we pair up high-entropy examples with their low-entropy sub-category counterparts in a label-aware fashion (E.g. easy high-entropy examples with easy low-entropy examples that have the same label). We empirically show that our training scheme leads to better overall and generalization performance without the need for any fancy optimization or auxiliary data. We report an absolute gain of 2% on the challenging iSarcasm dataset attesting to the potency of our technique.

In this report, we provide a thorough literature review of the research that has been done in relation to our problem statement. We also provide details of our training pipeline along with ablation experiments. Lastly, we conduct a thorough qualitative and quantitative error analysis. Based on its insights, we posit possible improvements and future directions.

2 Literature Survey

2.1 Dataset Quality Estimation

Training dynamics-based approaches are popular in machine learning literature to assess dataset difficulty and quality. Pleiss et al. (2020) proposed Area Under the Margin (AUM) statistic to distinguish correct vs. mislabelled samples. It measures the average difference between the logit values for an example's assigned label and its highest non-assigned label. Correctly labeled examples exhibit higher AUM values, whereas the hard and mislabelled examples are identified using an empirical threshold. Swayamdipta et al. (2020) extends this idea to identify easy-to-learn, ambiguous, and hard-to-learn examples. For each example, they calculate confidence and variability based on the mean and standard deviation of the gold label probabilities. The *easy-to-learn*, *ambiguous*, and *hard-to-learn* examples are categorized as examples with high confidence + low variability, high variability, and low confidence + low variability, respectively. Toneva et al. (2019) utilized training dynamics to identify frequently misclassified examples during later training epochs, despite being classified correctly during initial epochs. Their experiments reveal interesting findings: (i) specific examples are forgotten with high frequency, whereas some are never forgotten; (ii) The un-forgettable examples generalize across neural architectures; and (iii) A significant fraction of examples do not contribute much to the model's learnability and can be omitted during training while still maintaining state-of-the-art generalization performance. Studying training dynamics through the lens of identifying important examples, Paul et al. (2021) present two novel metrics, namely Gradient Normed (GraNd) and the Error L2-Norm (EL2N) that successfully identify important and difficult examples early in training. They further attest that pruning the training dataset with small GraNd does not sacrifice test performance. Adila and Kang (2022) use training dynamics to identify out-of-distribution samples.

More recently, Chiang and Lee (2022) investigated input representations, hidden layer representations, and output probabilities to distinguish in-distribution (ID), out-of-distribution (OOD), and adversarial examples. Their findings reveal that OOD examples exhibit differences from the initial layers themselves, while adversarial examples yield output probabilities with low confidence. Xu et al. (2021) provide an effective transformer approach to identify OOD examples using unsupervised in-domain data. On another note, Adila and Kang (2022) focuses on identifying dataset difficulty and difficult examples based on V-usable information (Xu et al., 2020). They present pointwise V-information (PVI) for measuring the difficulty of individual instances w.r.t. a given distribution. Finally, Mindermann et al. (2022) prioritize finding examples that are learnable, worth learning, and yet to learn using a probabilistic modeling selection function.

2.2 Mixup-based Data Augmentation

Mixup (Zhang et al., 2018) is a popular technique for data augmentation where additional samples are generated using a combination of different training examples and their labels. (Guo et al., 2019) presented a study of mixup-based interpolation of the word and sentence embeddings for text classification. Guo (2020) proffered a novel non-linear mixup technique where the mixing policy for the labels is adaptively learned based on the mixed input. Sun et al. (2020a) proposed a *Mixup-Transformer* with a stack of a mixup layer over the final hidden layer of the pre-trained transformer-based model

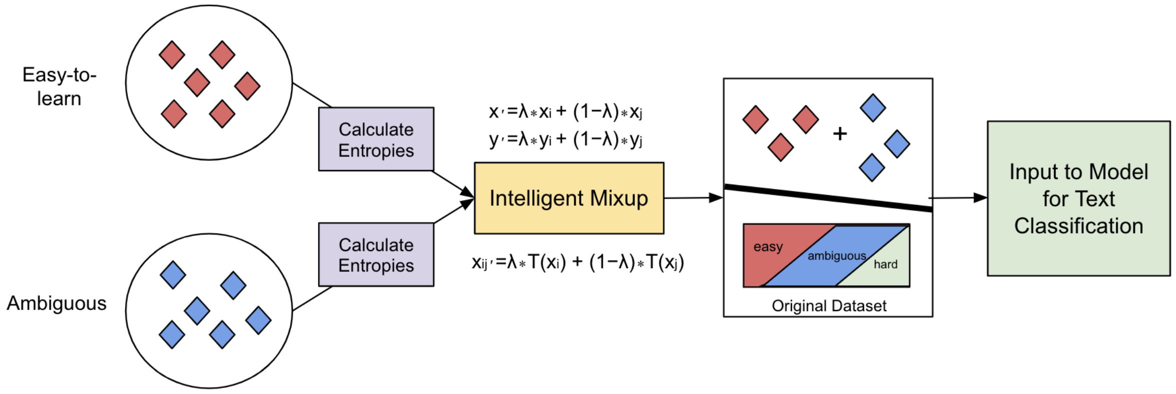
Easy-to-learn
Ambiguous
Calculate Entropies
Calculate Entropies
$x' = \lambda * x_i + (1-\lambda) * x_j$
$y' = \lambda * y_i + (1-\lambda) * y_j$
Intelligent Mixup
$x_{ij}' = \lambda * T(x_i) + (1-\lambda) * T(x_j)$
+
easy
ambiguous
hard
Original Dataset
Input to Model for Text Classification

Figure 1: Intelligent Mixup Technique

to do hidden space interpolation. Yin et al. (2021) come up with *Batch-Mixup*, which interpolates hidden states of the entire mini-batch, generating new points scattered throughout the space of the corresponding mini-batch. Guo (2020) presented a nonlinear mixup for out-Of-manifold data augmentation. TMix creates a large number of augmented training samples by interpolating text in hidden space (Chen et al., 2020). More recently, Yoon et al. (2021) proposed a mixup approach that does linear interpolation based on the saliency of tokens in the text. Lastly, Chen et al. (2022) proposed a double mixup strategy that uses the perturbed data and original data to carry out a two-step interpolation in the hidden space.

Drawing insights from these three approaches, we propose a unified framework that identifies a different variety of data points followed by an intelligent mixup.

3 Methodology

Our methodology consists of a two-stage pipeline starting with data quality estimation and followed by an intelligent mixup technique, the details of which are elaborated below.

3.1 Data Quality Estimation

Based on the experiments of Swayamdipta et al. (2020), we propose to unearth examples that are *easy-to-learn* and *ambiguous* and differentiate them from the *hard-to-learn* ones. The *easy-to-learn* examples have high confidence and low variability, thus, helping the model to converge faster. On the other hand, the *ambiguous* data points exhibit high variability, meaning that the model is often confused about their prediction. These examples help improve generalization and robustness. Lastly, the *hard-to-learn* examples are potentially mislabeled. We hypothesize that training with the pruned dataset containing *easy-to-learn* and *ambiguous* examples leads to better performance than training on the entire dataset. We use this as our backbone for further investigations.

Data cartography involves using data and mapping techniques to understand and visualize complex information. In natural language processing (NLP), data cartography can be used to analyze and visualize the relationships between different linguistic elements, such as words and sentences.

We apply this technique by creating maps of word co-occurrence. These maps show how often different words appear together in a text corpus, allowing us to identify patterns and relationships between words.

Data cartography is also be used to visualize the relationships between different texts or documents. For example, a network diagram can show how texts are related to each other based on shared words or concepts. This can be useful for identifying clusters of similar texts or for understanding the overall structure of a corpus of texts. It also helps us segregate the training data on the basis of the complexity to be learned by the model. Inspired by the work of Swayamdipta et al. (2020), we use this technique to classify mislabelled datapoints and segregate the other points as easy-to-learn and hard-to-learn. For this, we generate the training dynamics for every data point for every epoch. Using these logits, we filter out the different categories of training examples and train our model to observe its performance individually on each of the easy-to-learn, ambiguous data points and a combination of these.

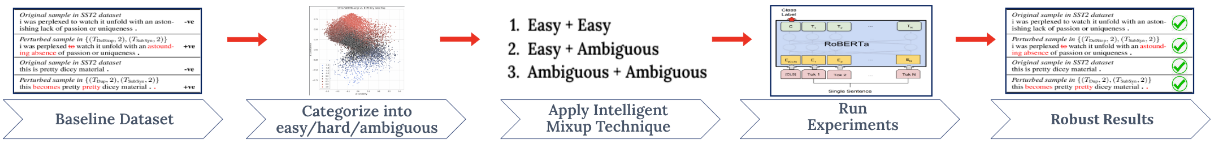
Original sample in SST2 dataset
i was perplexed to watch it unfold with an astonishing lack of passion or uniqueness .
-ve
Perturbed sample in {(T_DelStop, 2), (T_SubSyn, 2)}
i was perplexed to watch it unfold with an astounding absence of passion or uniqueness .
+ve
Original sample in SST2 dataset
this is pretty dicey material .
-ve
Perturbed sample in {(T_Dup, 2), (T_SubSyn, 2)}
this becomes pretty pretty dicey material . .
+ve
Baseline Dataset
Categorize into easy/hard/ambiguous
1. Easy + Easy
2. Easy + Ambiguous
3. Ambiguous + Ambiguous
Apply Intelligent Mixup Technique
Class Label
RoBERTa
Single Sentence
Run Experiments
Robust Results

Figure 2: Training pipeline

3.2 Intelligent Mixup Technique

Mixup (Zhang et al., 2018) is a data augmentation technique that produces synthetic examples via linear interpolation of two or more data points. As the interpolation happens in the linear space, the newly generated data lies in the same manifold as the original data points. Thus, mixup helps generate diverse data points valid to the original data distribution. Mixup also has a regularizing effect, as it prevents the model from overfitting on the original training data and makes it robust to small linear perturbations.

Mathematically, we can formulate a mixup as follows. Given two datapoints x_i, x_j with their respective labels y_i, y_j; mixup generates synthetic data x'_{ij} and its label y'_{ij} via linear interpolation as given in equation 1, 2. Here λ is the mixup ratio which can be either be a fixed value in $[0, 1]$ or $\lambda \sim \text{Beta}(\alpha, \alpha)$, for $\alpha \in (0, \infty)$.

$$x'_{ij} = \lambda * x_i + (1 - \lambda) * x_j \tag{1}$$

$$y'_{ij} = \lambda * y_i + (1 - \lambda) * y_j \tag{2}$$

For interpolation in the latent sub-space, x'_{ij} is generated as given in equation 3. Here $T(x_i)$ and $T(x_j)$ are output features of x_i, x_j from the transformer model.

$$x'_{ij} = \lambda * T(x_i) + (1 - \lambda) * T(x_j) \tag{3}$$

In a traditional mixup setup, x_i and x_j are selected randomly. This can lead to a high likelihood of simple points being paired together on which the model is already confident. This will provide little to no new information to the model. Thus, random sampling gives us little control over the generated synthetic examples. Therefore, we use an intelligent mixup technique that takes inspiration from the data maps. We facilitate the pairing of points from the same data map category. More specifically, we pair *easy-to-learn* examples with *easy-to-learn* ones and *ambiguous* examples with *ambiguous* ones. Such pairing helps the model generate examples from the same underlying distribution. We ignore the *hard-to-learn* examples as they often constitute mislabeled instances.

Moreover, we also utilize the entropy information calculated while obtaining the data maps. Entropy measures how much information an example provides in the learning process. Examples with low entropy have less information as the model can already predict its label confidently. On the other hand, examples with high entropy are more valuable in the learning process as they contain more information. For each data map category (easy and ambiguous), we pair the high entropy examples with the low entropy ones so that the generated synthetic data does not only have too easy and too difficult examples. Furthermore, we do this pairing in a label-aware fashion, i.e., examples of the positive class are paired with the positive class, while the negative ones are paired with other negative examples. This three-step pairing involving data maps category, entropy, and label information provides better results than random sampling, as shown in our experiments. Thus, we generate synthetic examples using this apriori pairing method. Furthermore, we add a dense layer, followed by GeLU activation (Hendrycks and Gimpel, 2016), layer normalization (Ba et al., 2016), and dropout (Srivastava et al., 2014) to add a learnable component on top of the mixup representations. Empirically, we found GeLU to give better and more stable results than ELU (Clevert et al., 2015), ReLU (Nair and Hinton, 2010), and Leaky ReLU.

4 Experimental Setup

4.1 Datasets

The commonly used text classification datasets of IMDB, Yelp Reviews, and Stanford Sentiment Treebank, are large in size and report excellent results by simple fine-tuning. Thus, we experiment with complex niche corpora for natural language understanding tasks and have limited training data. We use the challenging iSarcasm dataset (Oprea and Magdy, 2020), wherein the task is to identify whether a tweet is sarcastic. Different from the prevalent sarcasm identification datasets, tweets in iSarcasm corpus are labeled for sarcasm directly by their authors. Thus the name *intended-sarcasm*.

Dataset	Train	Test	Task Information
iSarcasm	4002	1617	Sarcasm Detection
Disaster	6089	1727	Disaster Tweet Classification

Table 1: Dataset Information

Moreover, examining the state-of-the-art sarcasm detection models on iSarcasm showed low performance compared to previously studied datasets (Oprea and Magdy, 2020). We also show results a real-time data corpora of disaster tweets from a local source in Kaggle [1]. Since this data was not validated yet represents a very emminent real-time problem, we chose this data for our experimentation. Information about both datasets are given in Table 1.

4.2 Baseline Methods

We experiment with the methods listed below. ▷ **Only Easy**: Train the text classification model on only the *easy-to-learn* examples. ▷ **Only Ambiguous**: Train using only the *ambiguous* examples. ▷ **Easy+Ambiguous**: Train on the combined corpus of *easy-to-learn* and *ambiguous* examples. ▷ **Random 33%**: As *easy-to-learn* and *ambiguous* subsets correspond to 33% of the entire dataset, we also experiment with a random subsample of 33%. ▷ **Full Corpus**: Train with the original dataset. ▷ **Full - mislabeled**: Train excluding only the *hard-to-learn* examples as they are the potentially mislabeled ones.

4.3 Training and Hyperparamter Setting

We experiment with all the models mentioned in the section 4.2. Each text is tokenized and padded/truncated to the maximum text length in the training corpus. We choose the *roberta-base* (Liu et al., 2019) version for all the experiments. After a few experiments, we narrowed down on the following hyperparameters: Adam optimizer with decoupled weight decay regularization (Loshchilov and Hutter, 2019), batch size of 8, a learning rate of $1e-5$, weight decay of $1e-2$, and dropout probability of 0.25. The mixup ratio λ is randomly sampled at each step from a Beta distribution with $\alpha = 1.0$. The models were evaluated using micro-f1. The models were trained for a maximum of 10 epochs with an early stopping patience of 5 if the validation micro-f1 does not improve. Since we use latent space-based interpolation, we can dynamically activate the mixup during the training. Similar to Sun et al. (2020b), we fine-tune the model without mixup in the first four epochs for good representations and add mixup for the rest of the epochs. All the models are trained using the binary cross entropy loss. The coding is done using PyTorch and HuggingFace.

5 Results and Discussions

Table 2 presents our results for the iSarcasm and Disaster dataset. As illustrated in the table, we see that using *easy* + *ambi* data from the entire dataset gives comparable or even better results than using the entire dataset *100%*. This shows that even with less amount but quality data, we are able to reach the same performance as the original dataset. We observed that using only the *ambiguous* examples leads to the worst performance across all the metrics. This can be attributed to the fact that these examples have high variability, and the model is inherently not confident about its predictions. Thus, training using only those examples does not help the model to learn and converge. This can also be observed from Figures 3 and 4. Using *easy* examples, we get significantly better results. However, they are still behind the full training scores, thus, suggesting that a good blend of *ambiguous* and *easy* is required. Training using randomly sampled 33% of the data and training with *ambiguous* and *easy* leads to similar performances, with random sampling outperforming by about 2% in f1-score for both the datasets. Removing the *hard-to-learn*, i.e., mislabeled examples, improves the scores marginally for both datasets, again showing the importance of quality data over quantity.

For mixup, we see that the results for the iSarcasm and Disaster dataset show a positive trend. Traditional mixup technique that randomly samples examples for linear interpolation reports almost similar results to using the entire dataset. The ratio of precision and recall, however, is swapped. Data maps-based mixup leads to an improvement of an avergae of 2% in f1 and 4% in the recall, with the precision dropping by 1%. Doing a label-aware mixup, we get better results with f1 increasing by 2%, recall by 1.5%, precision by 3.5%, and accuracy by 2% over traditional mixup. Finally, a label+entropy aware mixup leads to even more gains, with an average of 3%, 1.5%, 4%, and 2.5% increases in f1, recall, precision, and accuracy, respectively.

[1] https://www.kaggle.com/competitions/nlp-getting-started/data?select=train.csv

Datsets		100%	without-mixup					with-mixup			
			random 33%	easy	ambi	easy + ambi	full - hard	random	ambi + easy	ambi + easy + label	ambi + easy + entropy
Sarcasm	Accuracy	81.35	77.42	**85.71**	36.71	73.42	84.35	83.07	83.28	85.14	**85.57**
	Precision	64.18	57.49	42.85	51.56	56.43	**67.11**	65.32	66.81	68.99	**69.88**
	Recall	**66.83**	58.91	50.00	52.66	59.50	64.83	65.13	**68.58**	66.54	66.79
	F1	**65.25**	58.01	46.15	35.20	56.83	65.82	65.22	67.60	67.61	**68.11**
Disaster	Accuracy	**83.12**	80.73	79.05	63.69	81.22	82.99	83.45	83.19	82.99	**84.11**
	Precision	**83.35**	82.26	78.64	63.28	80.63	83.15	83.35	84.34	83.12	**84.67**
	Recall	81.56	**81.94**	79.46	63.68	80.54	81.47	81.57	81.48	81.91	**82.54**
	F1	82.16	82.09	78.76	63.22	80.58	82.05	82.16	82.30	82.36	**83.05**

Table 2: Results for Sarcasm and Disaster Tweets dataset with and without different flavors of the mixup technique

	Instance	Example	Gold-Label
Sarcasm	easy	Seeing John Mulaney perform tonight might honestly be one of my favorite experiences I've ever had!	0
	hard	who thinks me and sam should start a band called 'teeth at midnight'.	1
	ambiguous	Going out is overrated! On the couch watching serial killer documentaries is how I like to spend my Fridays.	0
Disaster	easy	Just got sent this photo from Ruby #Alaska as smoke from #wildfires pours into a school.	1
	hard	I was over here dreaming peacefully then that loud ass thunder wanted to scare me?	1
	ambiguous	I hate this damn Milwaukee IndyFest. All the cars sound like a really long tornado siren going off and it woke me up from my nap.	0

Table 4: Examples of easy to learn, hard to learn and ambiguous intsances from Sarcasm and Disaster dataset.

6 Error Analysis

6.1 Training Accuracy v/s Epoch

From the figures 3 and 4, we observe that training on ambiguous datasets leads to a worsened performance since the model doesn't converge even after 4 epochs. On the other hand, a mix of easy and ambiguous help the model generalize well over both categories and the model converges much faster. Since the random dataset also has a good mix of both easy and ambiguous examples, that model also converges pretty quickly. Another interesting observation is that easy+ambiguous mix of data converges much faster than random, since the model is now not confused by the examples on which it was actually going wrong. Removing those data points from the set helps it learn much better and helps boost the performance.

6.2 F1 Score trend for Mixup Technique

We consider the F1 score of full dataset as the baseline to compare our mixup techniques. Given the baselines, in figures 5 and 6, we see a common trend for F1 scores on both datasets with respect to the mixup technique we use. Intelligent mixup performs much better than the random mixup for both datasets. While our intelligent mixup techniques perform well, considering entropy to select data points performs even better. This is because the points chosen with higher entropy imply the model is not confident enough in selecting the data point, while a lower entropy suggests the model is

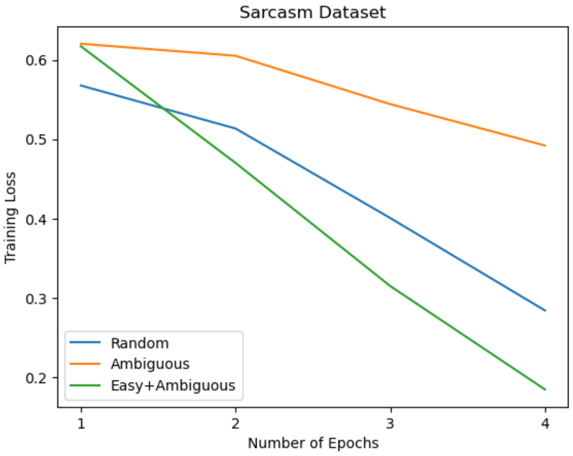

Figure 3: Training Accuracy v/s Epochs for Sarcasm Dataset

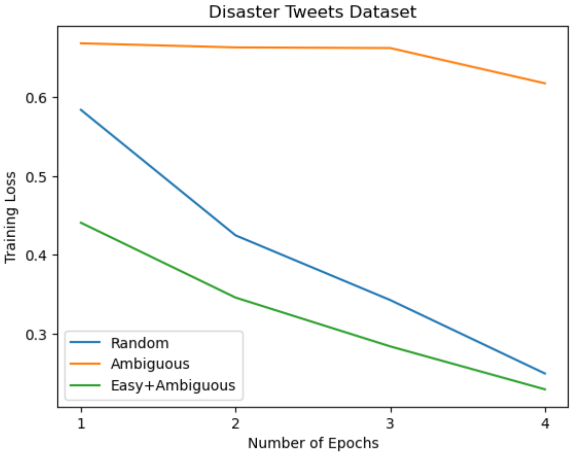

Figure 4: Training Accuracy v/s Epochs for Disaster Tweets Dataset

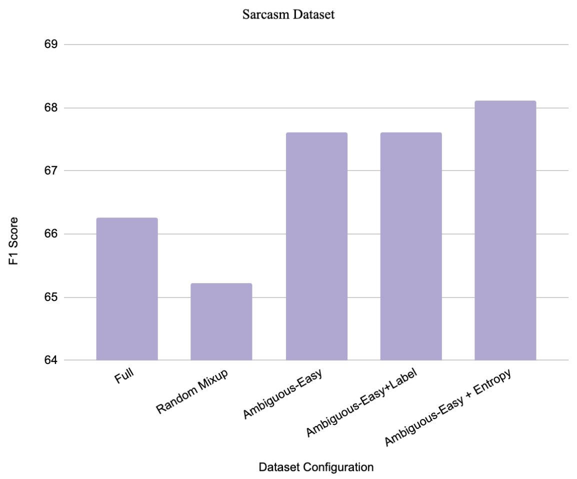

Figure 5: F1 Score for different mixup configurations for Sarcasm Dataset

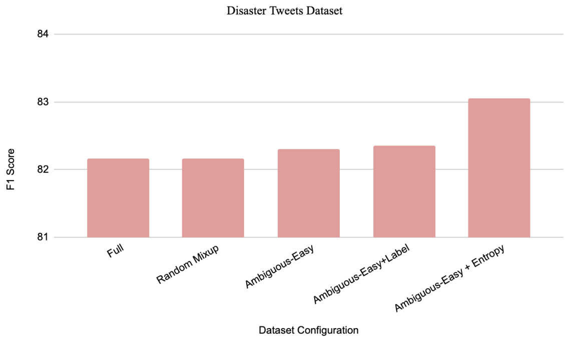

Figure 6: F1 Score for different mixup configurations for Disaster Tweets Dataset

confidently predicting the sample. For our mixup, we select our data points with a good mix of high and low entropy points, we get a better distribution in this case which helps boost the performance. On the other hand, a random mixup may not be useful because of the inability to compare the similarity between two samples in order to interpolate between them. In the case of text data using random mixup, it is often difficult to define a meaningful similarity metric, leading to a stagnation in performance after a particular amount of time.

6.3 Example-wise Analysis

We summarize a few examples from each of the two datasets (Disaster Tweets and iSarcasm dataset) to understand at the data level how the data mapping technique performs. As we observe, the examples in easy-to-learn category are straightforward in context having no word play within the meaning of the sentence. It is easy for the model to learn such sentences and overfit the model. However, adding ambiguous examples help to generalize the model as they are not as obvious in their semantics, which is primarily the reason it balances out the effect of easy-to-learn examples. For example in the sentence in the above table - *"Going out is overrated! On the couch watching serial killer documentaries is how I like to spend my Fridays."*, it is hard to understand if the speaker is serious or just throwing a sarcastic remark at someone else. Sentences like these, although predicted correctly, still confuse the model and prevent it from overfitting. However, there are still some examples on which the model is unable to predict correctly. Many a times these sentences are not only hard for the model but baffling for a human annotator. For such mislabelled examples like - *"I was over here dreaming peacefully then that loud ass thunder wanted to scare me?"*, we can easily notice that model is unable to make sense of words such as *"thunder"* that could help us determine that it is a disaster-related tweet. Poorly written tweets like these can confuse the human annotators too and should hence be removed from further experimentation to make the model more robust.

7 Conclusion and Future Work

Mixup-based latent interpolation can successfully generate synthetic examples when the amount of training data is less. In this work, we propose a training dynamics and dataset entropy-informed mixup for robust text classification. Intending to unearth data points that are difficult and easy to learn, we employ a training dynamics-based technique called '*Data Maps*' to identify *easy-to-learn*, *ambiguous*, and *hard-to-learn* examples in a given dataset. Then we propose an apriori datapoints matching scheme, wherein each datapoint is paired amongst other examples sharing the same label (sarcastic or non-sarcastic) and same category (easy or ambiguous). Moreover, the matching is done so that the examples with the highest entropy are paired with the lowest ones. We report results using different subsets of data and a varied combination of mixups. We report an impressive gain of more than 2% in the f1-score on the challenging iSarcasm dataset. We also see marginal gains on the disaster dataset. Our methodology indicates that we can gain promising results without extensive data augmentation if we utilize mixup in an intelligent way.

The future work of this research can take multiple directions. To begin with, the experiments should be scaled to other tasks, languages, and domains to attest to the proposed method's generalizability. A non-linear mixup technique similar to the one proposed by Guo (2020) could be utilized to increase coverage beyond the linear manifold. Lastly, curriculum learning techniques could be employed to efficiently sample data points.

References

Dyah Adila and Dongyeop Kang. 2022. Understanding out-of-distribution: A perspective of data dynamics. In *Proceedings on "I (Still) Can't Believe It's Not Better!" at NeurIPS 2021 Workshops*, volume 163 of *Proceedings of Machine Learning Research*, pages 1–8. PMLR.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 233–242. PMLR.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Hui Chen, Wei Han, Diyi Yang, and Soujanya Poria. 2022. DoubleMix: Simple interpolation-based data augmentation for text classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4622–4632, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, Online. Association for Computational Linguistics.

Cheng-Han Chiang and Hung-yi Lee. 2022. Understanding, detecting, and separating out-of-distribution samples and adversarial samples in text classification. *arXiv preprint arXiv:2204.04458*.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Hongyu Guo. 2020. Nonlinear mixup: Out-of-manifold data augmentation for text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):4044–4051.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Siddharth Karamcheti, Ranjay Krishna, Li Fei-Fei, and Christopher Manning. 2021. Mind your outliers! investigating the negative impact of outliers on active learning for visual question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7265–7281, Online. Association for Computational Linguistics.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Hassan H. Malik and Vikas S. Bhardwaj. 2011. Automatic training data cleaning for text classification. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 442–449.

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. 2022. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15630–15649. PMLR.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814.

Silviu Oprea and Walid Magdy. 2020. iSarcasm: A dataset of intended sarcasm. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1279–1289, Online. Association for Computational Linguistics.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. 2021. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems*, volume 34, pages 20596–20607. Curran Associates, Inc.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. 2020. Identifying mislabeled data using the area under the margin ranking. In *Advances in Neural Information Processing Systems*, volume 33, pages 17044–17056. Curran Associates, Inc.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip Yu, and Lifang He. 2020a. Mixup-transformer: Dynamic data augmentation for NLP tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3436–3440, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip S Yu, and Lifang He. 2020b. Mixup-transformer: dynamic data augmentation for nlp tasks. *arXiv preprint arXiv:2010.02394*.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. 2019. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Keyang Xu, Tongzheng Ren, Shikun Zhang, Yihao Feng, and Caiming Xiong. 2021. Unsupervised out-of-domain detection via pre-trained transformers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1052–1061, Online. Association for Computational Linguistics.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. 2020. A theory of usable information under computational constraints. In *International Conference on Learning Representations*.

Wenpeng Yin, Huan Wang, Jin Qu, and Caiming Xiong. 2021. BatchMixup: Improving training by interpolating hidden states of the entire mini-batch. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4908–4912, Online. Association for Computational Linguistics.

Soyoung Yoon, Gyuwan Kim, and Kyumin Park. 2021. SSMix: Saliency-based span mixup for text classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3225–3234, Online. Association for Computational Linguistics.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.