Handling Extreme Class Imbalance in Technical Logbook Datasets

Farhad Akhbardeh, Cecilia Ovesdotter Alm, Marcos Zampieri, Travis Desell
Rochester Institute of Technology
Rochester, NY, USA
{fa3019, coagla, mazgla, tjdvse}@rit.edu

Abstract

Technical logbooks are a challenging and under-explored text type in automated event identification. These texts are typically short and written in non-standard yet technical language, posing challenges to off-the-shelf NLP pipelines. The granularity of issue types described in these datasets additionally leads to class imbalance, making it challenging for models to accurately predict which issue each logbook entry describes. In this paper we focus on the problem of technical issue classification by considering logbook datasets from the automotive, aviation, and facilities maintenance domains. We adapt a feedback strategy from computer vision for handling extreme class imbalance, which resamples the training data based on its error in the prediction process. Our experiments show that with statistical significance this feedback strategy provides the best results for four different neural network models trained across a suite of seven different technical logbook datasets from distinct technical domains. The feedback strategy is also generic and could be applied to any learning problem with substantial class imbalances.

1 Introduction

Predictive maintenance techniques are applied to engineering systems to estimate when maintenance should be performed to reduce costs and improve operational efficiency (Carvalho et al., 2019), as well as mitigate risk and increase safety. Maintenance records are an important source of information for predictive maintenance (McArthur et al., 2018). These records are often stored in the form of technical logbooks in which each entry contains fields that identify and describe a maintenance issue (Akhbardeh et al., 2020a). Being able to classify these technical events is an important step in the development of predictive maintenance systems.

In most technical logbooks, issues are manually labeled by domain experts (*e.g.*, mechanics) in free text fields. This text can then be used to classify or cluster events by semantic similarity. Classifying events in technical logbooks is a challenging problem for the NLP community for several reasons: (a) the technical logbooks are written by various domain experts and contain short text entries with non-standard language including domain-specific abbreviated words (see Table 1 for examples), which makes them distinct from other short non-standard text corpora (*e.g.*, social media); (b) off-the-shelf NLP tools struggle to perform well on this type of data as they tend to be trained on standard contemporary corpora such as newspaper texts; (c) outside of the clinical and biomedical sciences, there is a lack of domain-specific, expert-based datasets for studying expert-based event classification, and in particular few resources are available for technical problem domains; and (d) technical logbooks tend to be characterized by a large number of event classes that are highly imbalanced.

Original Entry	**Pre-processed Entry**
fwd eng baff seeal needs resecured.	forward engine baffle seal needs resecured.
r/h eng #3 intake **gsk** leaking.	right engine number 3 intake gasket leaking.
bird struck on **p/w** at **twy**. bird **rmvd**.	bird struck on pilot window at taxiway. bird removed
location **rptd** as **nm** from **rwy aprch** end.	location reported as new mexico from runway approach end.

Table 1: Original and text-normalized example data instances illustrating that domain-specific terms (*baffle*), abbreviations (*gsk - gasket, eng - engine*), and misspellings (*seeal - seal*) are abundant in logbook data.

We address the aforementioned challenges with a special focus on exploring strategies to address class imbalance. There is wide variation in the number of instances among the technical event classes examined in this work, as shown in Figure 1 and Ta-

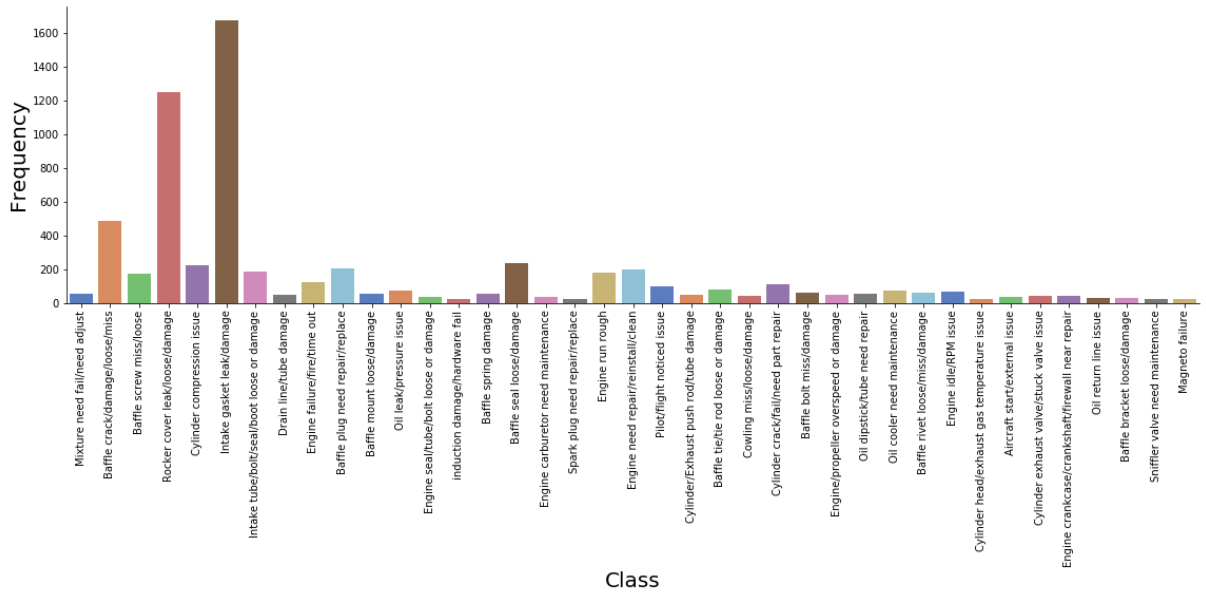

Figure 1: Number of instances in 39 unbalanced classes of the aviation maintenance (*Avi-Main*) dataset.

ble 3. This extreme class imbalance is an obstacle when processing logbooks as it causes most learning algorithms to become biased and mainly predict the large classes (Kim et al., 2019). To overcome this issue, we introduce a feedback loop strategy, which is a repurposing of a method used to address extreme class imbalance in computer vision (Bowley et al., 2019), and examine it for classification of textual technical event descriptions. This technique is applied in the training of a suite of common classification models on seven predictive maintenance datasets representing the aviation, automotive, and facility maintenance domains.

This paper addresses these research questions:
RQ1: To which extent does the class granularity and class imbalance present in technical logbooks impact technical event classification performance, and can a feedback loop for training data selection effectively address this issue?
RQ2: Which classification models are better suited to classify technical events for predictive maintenance across logbook datasets representing different technical domains?
The main contributions of this work include:

1. Experimental results showing strong performance of the feedback loop in addressing the class imbalance problem in technical event classification across all datasets and models;
2. A thorough empirical evaluation of the performance of the technical event classifier considering multiple models and seven logbook datasets from three different domains.

2 Related Work

Most expert-domain datasets containing events have focused on healthcare. For instance, Altuncu et al. (2019) analyzed patient incidents in unstructured electronic health records provided by the U.K. National Health Service. They evaluated a deep artificial neural network model on the expert-annotated textual dataset of a safety incident to identify similar events that occurred. Deléger et al. (2010) proposed a method to deal with unstructured clinical records, using rule-based techniques to extract names of medicines and related information such as prescribed dosage. Savova et al. (2010) considered free-text electronic medical records for information extraction purposes and developed a system to obtain clinical domain knowledge.

Patrick and Li (2009) proposed the cascade methods of extracting the medication records such as treatment duration or reason, obtained from patient's historical records. Their approach for event extraction includes text normalization, tokenization, and context identification. A system using multiple features outperformed a baseline method using a bag of words model. Yetisgen-Yildiz et al. (2013) proposed the lung disease phenotypes identification method to prevent the use of a hand-operated identification strategy. They employed NLP pipelines including text pre-processing and further text classification on the textual reports to identify the patients with a positive diagnosis for the disease. Based on the outcome, they achieve

Tech. Event or Issue Label	Example Instance of Technical Logbook Entry	Abbr., Misspelling, Terminology
SUBSTANTIAL DAMAGE	(1) **AFT** ON TAXI, **WING STRUECK FUEL** TRUCK, CHANDLER, AZ	AFT, WING, STRUECK, FUEL
BAFFLE DAMAGE	(2) **R/H FWD** UPPER **BAFF** SEAL NEEDS TO BE RESECURED	R/H, FWD, BAFL
MINOR DAMAGE	(1) SAW **SML** FLOCK FLYING UPON **LDG** FLARE, ACROSS **RWY**	SML, LDG, RWY
UNKNOWN	(1) NO **DMG**. BIRD REMAINS ON **F/O WINDSCREEN**	DMG, F/O, WINDSCREEN
PM SERVICE	(3) **PM** SERVICES CHECK **TIRES** FOR LEAKS CHECK **PLOW BATT**	PM,TIRES, PLOW, BATT
DRIVING ISSUE	(4) FAILURE TO YIELD RIGHT, **OVE** CORRECTING OVER **STEERING**	OVE, STEERING
STOP SIGN RUNNING	(4) MOTORISTS REGULARLY ILLEGAL **U-TURNS** IN **R/HOUR**	U-TURNS, R/HOUR
BUILDING PM	(5) THE **A/C** UNIT IN THE KITCHEN ON 3TH FLOOR **DMG**/LEAK	A/C, DMG
ENG NEED REPAIR	(3) CHANGE **OIL** & FILTER: **L/H ENG**, CHECK **COMP** & **PLUGS**	OIL, ENG, L/H, COMP, PLUGS
PREVENTIVE MAINT	(5) RESET **BOILER** #2 **TMER**, CHECKED **BLDG**. THROUGHOUT	BOILER, BLDG

Table 2: Example instances of technical logbook entries spanning the aviation accident (1), aviation maintenance (2), automotive maintenance (3), automotive safety (4), and facility maintenance (5). Each instance shows how domain-specific terminology, abbreviations (Abbr.), and misspelled words (in bold font) are used by the domain expert, and also illustrates some of the event types covered. More details are provided in Section 3.

notable performance by using the n-gram features with the Maximum Entropy (MaxEnt) classifier.

There is also relevant research on event classification in social media. For example, Ritter et al. (2012) proposed an open-source event extraction and supervised tagger for noisy microblogs. Cherry and Guo (2015) applied word embedding-based modeling for information extraction on news-wire and tweets, comparing named entity taggers to improve their method. Hammar et al. (2018) performed experimental work on Instagram text using weakly supervised text classification to extracted clothing brand based on user descriptions in posts.

The problem of class imbalance has been studied in recent years for numerous natural language processing tasks. Tayyar Madabushi et al. (2019) studied automatic propaganda event detection from a news dataset using a pre-trained BERT model. They recognized that the BERT model had issues in generalizing. To overcome this issue, they proposed a cost-weighting method. Al-Azani and El-Alfy (2017) analyzed polarity measurement in imbalanced tweet datasets utilizing features learned with word embeddings. Li and Nenkova (2014) studied the class imbalance problem in the task of discourse relation identification by comparing the accuracy of multiple classifiers. They showed that utilizing a unified method and further down-sampling the negative instances can significantly enhance the performance of the prediction model on unbalanced binary and multi-classes.

Dealing with unbalance classes is also studied well in the sentiment classification task. Li et al. (2012) introduced an active learning method that overcomes the problem of data class unbalance by choosing the significant sample of minority class for manual annotation and majority class for automatic annotation to lower the amount of human annotation required. Furthermore, Damaschk et al. (2019) examined techniques to overcome the problem of dealing with high-class imbalance in classifying a collection of song lyrics. They employed neural network models including a multi-layer perceptron and a Doc2Vec model in their experiments where the finding was that undersampling the majority class can be a reasonable approach to remove the data sparsity and further improve the classification performance.

Li et al. (2020) also explored the problem of high data imbalance using cross-entropy criteria as well as standard performance metrics. They proposed a loss function called Dice loss that assigns equal importance to the false negatives and the false positives. In computer vision, Bowley et al. (2019) developed an automated feedback loop method to identify and classify wildlife species from Unmanned Aerial Systems imagery, for training CNNs to overcome the unbalanced class issue. On their expert imagery dataset, the error rate decreased substantially from 0.88 to 0.05. This work adapts this feedback loop strategy to the NLP problem of classifying technical events.

3 Technical Event Datasets

In this work, we used a set of 7 logbook datasets from the aviation, automotive, and facility domains available at MaintNet (Akhbardeh et al., 2020a). MaintNet is a collaborative open-source platform for predictive maintenance language resources featuring multiple technical logbook datasets and tools. These datasets include: 1) *Avi-Main* contains seven years of maintenance logbook reports collected by

Code	Inst	Avg Toks	N Cls	Class Size			
				Min	Med	Avg	Max
Avi-Main	6,169	13.85	39	21	56	158	1,674
Avi-Acc	4,130	14.31	5	179	966	826	1,595
Avi-Safe	17,718	19.52	2	2,134	8,859	8,859	15,584
Auto-Main	617	7.34	5	23	48	123	268
Auto-Acc	52,707	4.59	3	1,085	11,060	17,569	40,562
Auto-Safe	4,824	25.11	17	86	213	284	678
Faci-Main	74,360	31.50	70	25	303	1,062	10,748

Table 3: Number of instances (Inst), average number of tokens per instance (Avg Toks), number of classes (N Cls), and class size statistics: minimum, average, median, and maximum (Min, Med, Avg, Max) for each dataset.

the University of North Dakota aviation program on aircraft maintenance that were reported by the mechanic or pilot. 2) *Avi-Acc* contains four years of aviation accident and reported damages. 3) *Avi-Safe* contains eleven years of aviation safety and incident reports. Accidents were caused by foreign objects/birds during the flights which led to safety inspection and maintenance, where safety crews indicated the damage (safety) level for further analysis. 4) *Auto-Main* is a single year report with maintenance records for cars. 5) *Auto-Acc* contains twelve years of car accidents and crash reports describing the related car maintenance issue and property damaged in the accident. 6) *Auto-Safe* contains four years of noted hazards and incidents on the roadway from the driver. 7) *Faci-Main* contains six years of logbook reports collected for building maintenance.

These technical logbooks include short, compact, and descriptive domain-specific English texts single instances usually contain between 2 and 20 tokens on average including abbreviations and domain-specific words. An example instance from Table 2, *r/h fwd upper baff seal needs to be resecured*, shows how the instances for a specific issue class are comprised from specific vocabulary (less ambiguity), and therefore contain a high level of granularity (level of description for an event from multiple words) (Mulkar-Mehta et al., 2011). Table 3 presents statistics for each dataset, in terms of the number of instances, average instance length, number of classes, and the minimum, average, median and maximum class size to represent how imbalanced the datasets are.

An instance in the logbook can be formed as a complete description of the technical event (such as a safety or maintenance inspection) like: *#2 & #4 cyl rocker cover gsk are leaking*, or it might contain an incomplete description by solely referring to the damaged part/section of machinery (*hyd cap chck eng light on*) using few domain words. In either form of the problem description, the given annotation (label) is at the issue type-level, *e.g.*, *baffle damage*. Table 2 shows multiple examples with associated instances.

Further characteristics of these log entries include compound words (*antifreeze, engine-holder, driftangle, dashboard*). Many of these words (*e.g.*, a compound word: *dashboard*) essentially represent the items, or domain-specific parts used in the descriptions. Additionally, function words (*e.g.*, prepositions) are important and removing them could alter the meaning of the entry. The logbook datasets also have both the following shared and distinct characteristics:

Shared Characteristics: Each instance contains a descriptive observation of the issue and/or the suggested action that should be taken (*eng inspection panel missing screw*). Each instance also refers to maintaining a single event, which means the recognized problem applies to the only single-issue type. As an example, the instance *cyl #1 baff cracked at screw support & forward baff below #1* includes a combination of sequences that refers to the location and/or specific part of the machinery.

Distinct Characteristics: In each domain, terminologies, a list of terms, and abbreviations are distinct, and an abbreviation can have different expansion depending on the domain context (Sproat et al., 2001), *e.g.*, *a/c* can mean *aircraft* in aviation and in the automotive domain *air conditioner*. However, the abbreviations and acronyms of the domain words (e.g. *atc - air traffic control*) in these technical datasets should not be approached as a word sense disambiguation problem as they require character level expansion.

4 Methods and Models

4.1 Handling Class Imbalance

Collecting additional data to augment datasets is a common approach for tackling the problem of skewed class distributions. However, as discussed earlier, technical logbooks are proprietary and very hard to obtain. In addition, each domain captures domain-specific lexical semantics, preventing the use of techniques such as domain adaption (Ma

Algorithm 1 Feedback Loop Pseudocode

▷ Gets $\mathcal{MCS}$ random instances from each class
function SAMPLERANDOM($\mathcal{C}$, $\mathcal{MCS}$)
  Array $\mathcal{A}$
  for $i \leftarrow 1$ to SIZE($\mathcal{C}$) **do**
    SHUFFLE($\mathcal{C}_i$)
    $\mathcal{A} \leftarrow \mathcal{A} \cup$ GETFIRSTN($\mathcal{MCS}, \mathcal{C}_i$)
  return $\mathcal{A}$

▷ Gets $\mathcal{MCS}$ instances from each class with the worst error
function RESAMPLE($\mathcal{C}$, $\mathcal{M}$, $\mathcal{MCS}$)
  Array $\mathcal{A}$
  for $i \leftarrow 1$ to SIZE($\mathcal{C}$) **do**
    CALCULATEERROR($\mathcal{C}_i$)
    SORTBYERROR($\mathcal{C}_i$)
    $\mathcal{A} \leftarrow \mathcal{A} \cup$ GETFIRSTN($\mathcal{MCS}, \mathcal{C}_i$)
  return $\mathcal{A}$

Input: Training Data $\mathcal{D} = \text{Instance}(1, 2, \ldots, N)$
Input: Feedback Loop Iterations $\mathcal{FLI}$
Input: Epochs Per Loop Iteration $\mathcal{FLE}$
Input: Minimum Class Size $\mathcal{MCS}$

▷ Divide training data by class
Array $\mathcal{C} \leftarrow$ SPLITBYCLASS($\mathcal{D}$)

▷ Get initial active training data $\mathcal{A}$ randomly
Array $\mathcal{A} \leftarrow$ SAMPLERANDOM($\mathcal{C}$, $\mathcal{MCS}$)
Model $\mathcal{M}$
for $l \leftarrow 1$ to $\mathcal{FLI}$ **do**
  ▷ Train the model for the number of epochs per iteration
  $\mathcal{M} \leftarrow$ TRAIN($\mathcal{M}$, $\mathcal{FLE}$, $\mathcal{A}$)
  ▷ Update the active training data
  $\mathcal{A} \leftarrow$ RESAMPLE($\mathcal{D}$, $\mathcal{M}$, $\mathcal{MCS}$)

Output: $\mathcal{M}$

et al., 2019) to apply a large class data from one technical domain to another. For example, instances that describe an *engine failure* in the aviation domain are distinct from *engine failure* instances reported in the automotive domain. In this paper we apply five different methods for selecting training data for the models to analyze their effects on classification performance: (1) under(down)- and (2) over-sampling, (3) random down-sampling, (4) a feedback loop strategy, and (5) a baseline strategy which simply uses all available data.

Re-sampling Under- and over-sampling are resampling techniques (Maragoudakis et al., 2006) that were used to create balanced class sizes for model training. For over-sampling, instances of the minority classes are randomly copied so that all classes would have the same number of instances as the largest class. For under-sampling, observations are randomly removed from the majority classes, so that all classes have the same number of instances as the smallest class. For both approaches, we first divided our datasets into test and training sets before performing over-sampling to prevent contamination of the test set by having the same observations in both the training and test data.

Feedback Loop To address class imbalances in text classification, this work adapts the approach in Bowley et al. (2019) from the computer vision domain. The goal of this approach is not only to alleviate the bias towards majority classes but also to adjust the training data instances such that the models are always being trained on the instances that was performing the worst on. It should be noted that this approach is very similar to *adaptive learning* strategies which have been shown to aid in human learning (Kerr, 2015; Midgley, 2014).

Algorithm 1 presents pseudocode for the feedback loop. In this process, the active training data (the data used to actually train the models in each iteration of the loop) is continually resampled from the training data. The model is first initially trained with an undersampled number of random instances from each class, which becomes the initial active training data. The model $\mathcal{M}$ then performs inference over the entire training set, and then selects $\mathcal{MCS}$ instances from each class $\mathcal{C}_\text{i}$ which had the worst error during inference, where $\mathcal{MCS}$ is the minority (smallest) class size. The model is then retrained with this new active training data and the process of training, inference and selection of the $\mathcal{MCS}$ worst instances repeats for a fixed number of feedback loop iterations, $\mathcal{FLI}$. In this way the model is always being trained on the instances it has classified the worst.

To measure the effect of resampling the worst performing instances, the feedback loop approach was also compared to a random downsampling (DS) loop, where instead of evaluating the model over each instance and selecting the worst performing instances, $\mathcal{MCS}$ instances from each class are instead randomly sampled. As performing inference over the entire training set adds overhead, a comparison to the random DS loop method would show if performing this inference is worth the performance cost over simple random resampling. This approach is the same as Algorithm 1 except that `SampleRandom` is used instead of `Resample` in the feedback loop. Section 4.3 describes how the number of training epochs and loop iterations were determined such that all the training data selection methods are given a fair evaluation with the same amount of computational time.

Evaluation Metrics For imbalanced datasets, simply using precision, recall or F1 score metrics for the entire datasets would not accurately reflect how well a model or method performs, as they emphasize the majority classes. To overcome this, alternative evaluation metrics to handle the class imbalance problem were used, as recommended by Banerjee et al. (2019). Specifically, we report the models performance based on precision, recall, and F1 score by utilizing a macro-average over all classes, as this gives every class equal weight, and hence reveals how well the models and training data selection strategies perform.

4.2 Model Architecture and Training

Different machine learning methods were considered for technical event/issue classification (e.g. engine failure, turbine failure). Each instance is an individual short logbook entry and contains approximately 2 to 20 tokens (12 words on average per instance including function words), as shown in Table 3.The methods used in this study were a Deep Neural Network (DNN) (Dernoncourt et al., 2017), a Long Short-Term Memory (LSTM) (Suzgun et al., 2019), recurrent neural network (RNN) (Pascanu et al., 2013), a Convolutional Neural Network (CNN) (Lin et al., 2018), and BERT (Devlin et al., 2019).

Deep Neural Network A deep artificial neural network (DNN), as described by Dernoncourt et al. (2017), can learn abstract representation and features of the input instances that would help to achieve better performance on predicting the issue type in the logbook dataset. The DNN used was a 3 layer, fully connected feed forward neural network with an input embedding layer of dimension 300 and equal size number of words followed by 2 dense layers with 512 hidden units with ReLU activation functions followed by a dropout layer. Finally, we added a fully connected dense layer with size equal to the number of classes, with a SoftMax activation function.

Long Short-Term Memory An LSTM RNN was also used to perform a sequence-to-label classification. As described by Suzgun et al. (2019) LSTM RNNs utilize several vector gates at each state to regulate the passing of data by the sequence which enhances the modeling of the long-term dependencies. We used a 3 layer LSTM model with a word embedding layer of dimension 300 and the equal size number of words followed by an LSTM layer with setting the number of hidden units equal to the embedding dimension, followed by a dropout layer. Finally, we added a fully connected layer with size equal to the number of classes, with a SoftMax activation function.

Convolutional Neural Network Convolutional neural networks (CNNs) have demonstrated exceptional success in NLP tasks such as document classification, language modeling, or machine translation (Lin et al., 2018). As Xu et al. (2020) described, CNN models can produce consistent performance when applied to the various text types such as short sequences. We evaluated a CNN architecture (Shen et al., 2018) with a convolutional layer, followed by batch normalization, ReLU, and a dropout layer, which was followed by a max-pooling layer. The model contained 300 convolutional filters with the size of 1 by n-gram length pooling with the size of 1 by the length of the input sequence, followed by concatenation layer, then finally connected to a fully connected dense layer, and an output layer equal to the size of the dataset class using a SoftMax activation function.

Bidirectional Encoder Representations We also evaluated using the pre-trained uncased Bidirectional Encoder Representations (BERT) for English (Devlin et al., 2019). We fine-tuned the model, and used a word piece based BERT tokenizer for the tokenization process and the *RandomSampler* and *SequentialSampler* for training and testing respectively. To better optimize this model, a schedule was created for the learning rate that decayed linearly from the initial learning rate we set in the optimizer to 0.

4.3 Experimental Settings

Datasets and Baselines First, the technical text pre-processing pipeline developed by Akhbardeh et al. (2020b) was applied, which comprises domain-specific noise entity removal, dictionary-based standardization, lexical normalization, part of speech tagging, and domain-specific lemmatization. We divided the datasets selecting randomly from each class independently to maintain a similar class size distribution, using 80% of the instances for training and 20% of the instances for testing data. For feature extraction, two methods were considered: a bag-of-word model (n-grams:1) (Pedregosa et al., 2011) and pre-trained 300 dimensional GloVe word embeddings (Pennington et al., 2014).

Hyperparameter and Tuning The coarse to fine learning (CFL) approach (Lee et al., 2018) was used to set parameters and hyperparameters for the DNN, LSTM, and CNN models. Experiments considered batch sizes of 32, 64, and 128, an initial learning rate ranging from 0.01 to 0.001 with a learning decay rate of 0.9, and dropout regularization in the range from 0.2 to 0.5 in all models, as well as ReLU and SoftMax activation functions (Nair and Hinton, 2010), categorical cross-entropy (Zhang and Sabuncu, 2018) as the loss function, and the Adam optimizer (Kingma and Ba, 2015) in the DNN, LSTM, CNN and BERT models. Based on experiments and network training accuracy, a batch size of 64 and drop out regularization of 0.3 was selected for model training.

Each model with each training data selection strategy was trained 20 times to generate results for each dataset. To ensure each training data selection strategy was fairly compared with a similar computational budget, the number of training epochs and loop iterations (if the strategy had a feedback or random downsampling loop) were adjusted so that the total number of training instances evaluations each model performed was the same. For each dataset, the number of forward and backward passes, 'T' for 100 epochs of the baseline strategy was used as the standard. As an example, Table 4 shows how many loop iterations, epochs per loop, and inference passes were done for each training data selection strategy on the *Auto-Safe* dataset. Given the differences between the min and max class sizes it was not possible to get exact matches but the strategies came as close as possible. We counted each inference pass for the feedback loop the same as a forward and backward training pass, which actually was a slight computational disadvantage for the feedback loop, as a forward and backward pass in training takes approximately 1x to 2x the time as an inference pass.

Dataset	L	EPL	LTI	INM	T
Baseline	1	100	3,859	0	385,900
Downsampling	1	329	1,173	0	385,917
Oversampling	1	42	9,214	0	386,988
Random DS Loop	33	10	1,173	0	387,090
Feedback Loop	25	10	1,173	3,859	389,725

Table 4: Details regarding different training process using the various methods for handling the unbalanced class in automotive safety (*Auto-Safe*) dataset with 17 total classes. Loop (L), Epochs Per Loop (EPL), Active Training instance Size (LTI), Inference for New Misclassified (INM) and Total Instances Evaluated (T).

5 Results

Table 5 shows a comparison between the baseline and the four different class balancing methods (over-sampling, under-sampling, the random downsampling (DS) loop and the feedback loop). Based on these outcomes, the feedback loop strategy almost entirely outperforms the other methods over all datasets and models, showing that performing inference over the training set and reselecting the training data from the worst performing instances does provide a benefit to the learning process. A plausible explanation is that this strategy does not introduce bias into the larger class and also does not effect the minority class size distribution. It also does not waste training time on instances the model has already well learned.

Table 5 also shows the empirical analysis of the four classification models, with the model and training data selection strategy providing the overall best results shown in bold and italics. Using technical text pre-processing techniques described in Section 4.3, and the feedback loop strategy described in Section 4.1, the precision, recall, and F1 score improved compared to the baseline performance. The CNN model outperformed the other algorithms with improved precision, recall, and F1 score for almost all datasets except for *Avi-Main*, where BERT had the similar results, and *Auto-Main* where CNN and BERT tied. This is interesting, given the current popularity of the BERT model, however it may be due to the substantial lexical, topical, and structural linguistic differences between the technical logbook data and the English corpus that BERT was pre-trained on.

Furthermore, we conducted the Mann-Whitney U-test of statistical significance by using the F1 scores of each of the 20 repeated experiments of the classification models, using the baseline and the feedback loop approach as the two different populations. The outcomes are shown in Table 6, with the differences being highly statistically significant.

6 Discussion

To investigate the optimal strategies for dealing with these imbalanced technical datasets, we studied various methods on how to process the data, extract features, and classify the type of event. Re-

Dataset	Model	Baseline (%)			Down Sampling (%)			Over Sampling (%)			Random DS Loop (%)			Feedback Loop (%)		
		Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
Avi-Main	DNN	0.90	0.89	0.89	0.67	0.78	0.70	0.90	0.90	0.90	0.90	0.90	0.90	**0.93**	**0.91**	**0.91**
	LSTM	0.84	0.85	0.84	0.81	0.83	0.81	0.85	0.84	0.84	0.84	0.84	0.84	**0.86**	**0.88**	**0.87**
	CNN	0.93	0.92	0.92	0.89	0.88	0.88	0.94	0.92	0.92	0.93	0.91	0.91	***0.95***	**0.94**	**0.94**
	BERT	0.93	0.93	0.93	0.85	0.86	0.85	0.94	0.94	0.94	0.94	0.93	0.93	***0.95***	***0.96***	***0.95***
Avi-Acc	DNN	0.47	0.44	0.43	0.35	0.45	0.35	0.48	***0.47***	0.47	0.50	0.44	0.46	***0.52***	0.45	***0.48***
	LSTM	0.38	0.37	0.37	0.35	0.35	0.35	0.39	**0.39**	0.39	0.38	**0.39**	0.38	**0.40**	**0.39**	**0.39**
	CNN	0.50	***0.49***	***0.49***	0.43	0.42	0.42	***0.52***	0.44	0.47	0.51	0.44	0.47	***0.52***	0.46	0.48
	BERT	0.48	0.42	0.44	0.41	0.40	0.40	0.50	0.44	0.46	0.50	0.44	0.46	**0.51**	**0.45**	**0.47**
Avi-Safe	DNN	0.43	0.41	0.41	0.36	0.36	0.36	0.50	0.50	0.50	0.50	0.49	0.49	**0.53**	**0.51**	**0.51**
	LSTM	0.47	0.46	0.46	0.43	0.42	0.42	**0.49**	**0.50**	**0.49**	0.48	0.46	0.47	**0.49**	**0.50**	**0.49**
	CNN	0.59	0.57	0.57	0.50	0.50	0.50	0.60	0.59	0.59	0.59	0.59	0.59	***0.62***	***0.61***	***0.61***
	BERT	0.50	0.50	0.50	0.44	0.46	0.44	0.54	0.54	0.54	0.53	0.53	0.53	**0.56**	**0.57**	**0.56**
Auto-Main	DNN	0.58	0.45	0.49	0.33	0.49	0.39	0.60	**0.55**	0.56	0.58	0.54	0.55	**0.61**	**0.55**	**0.57**
	LSTM	0.49	0.55	0.51	0.41	0.42	0.41	0.50	0.60	0.54	0.51	0.58	0.54	**0.53**	**0.61**	**0.55**
	CNN	0.61	0.61	0.61	0.53	0.53	0.53	0.64	***0.64***	***0.64***	0.63	**0.64**	0.63	***0.65***	***0.64***	***0.64***
	BERT	0.60	0.60	0.60	0.54	0.53	0.53	0.63	***0.64***	0.63	0.63	0.63	0.63	**0.64**	***0.64***	***0.64***
Auto-Acc	DNN	0.43	0.34	0.30	0.35	0.42	0.27	0.39	**0.42**	0.31	0.40	0.39	0.39	**0.48**	0.40	**0.40**
	LSTM	0.45	0.39	0.41	0.40	0.40	0.40	0.42	**0.41**	0.41	0.42	0.40	0.40	**0.48**	**0.41**	**0.44**
	CNN	0.46	0.43	0.44	0.44	0.41	0.42	0.49	0.50	0.49	0.50	0.51	0.50	**0.51**	***0.53***	***0.52***
	BERT	0.50	0.49	0.49	0.47	0.47	0.47	0.50	0.50	0.50	0.51	0.49	0.50	***0.52***	**0.51**	**0.51**
Auto-Safe	DNN	0.52	0.46	0.48	0.40	0.47	0.41	0.54	0.51	0.51	0.54	0.51	0.51	**0.55**	**0.52**	**0.53**
	LSTM	0.40	0.40	0.40	0.38	0.39	0.38	0.41	**0.42**	0.41	0.41	0.41	0.41	**0.43**	**0.42**	**0.42**
	CNN	0.59	0.58	0.58	0.52	0.51	0.51	0.59	***0.60***	0.59	0.59	0.59	0.59	***0.62***	***0.60***	***0.61***
	BERT	0.57	0.56	0.56	0.52	0.50	0.50	**0.58**	0.56	0.56	0.57	0.57	0.57	**0.58**	**0.59**	**0.59**
Faci-Main	DNN	0.57	0.48	0.50	0.33	0.40	0.34	0.56	0.48	0.50	0.57	0.50	0.53	**0.59**	**0.51**	**0.54**
	LSTM	0.56	**0.56**	0.56	0.53	0.52	0.52	0.59	0.55	0.56	0.59	**0.56**	0.57	**0.63**	**0.56**	**0.60**
	CNN	0.64	0.64	0.64	0.61	0.60	0.60	0.66	0.66	0.66	0.65	0.65	0.65	***0.69***	***0.67***	***0.68***
	BERT	0.63	0.64	0.63	0.60	0.60	0.60	0.65	0.64	0.64	0.64	0.65	0.64	**0.68**	***0.67***	**0.67**

Table 5: Comparison of results for the 7 datasets, for the baseline and four methods to address class imbalance for the four evaluated models (DNN, LSTM, CNN and BERT). Each model's macro average performance is shown as precision (Pre), recall (Rec) and F1 score. The best results over the training data selection strategies are shown in bold, and the best results over all models are additionally in italics.

garding the discussion provided in Section 3 about the nature of such a dataset, there are key challenges that effect the performance of employed algorithms. As discussed in Section 1, the extreme class imbalance observed in these technical datasets substantially affects learning algorithms' performance. To overcome this issue, we first explored oversampling and undersampling, which both result in balanced class sizes. Undersampling removed portions of dataset that could be important for certain technical events or issues, which resulted in underfitting and weak generalization for important classes. On the other hand, oversampling may introduce overfitting in the minority class, as some of the event types are very short tokens containing domain-specific words. Following this, to minimize the possibility of overfitting and underfitting, a random downsampling loop and a feedback loop were investigated to minimize bias in the training process. It was found that the added computational cost of the feedback loop inference was worth the reduction in training time it caused over the random downsampling loop.

The scarce data available in a dataset such as *Auto-Main* is certainly an issue for deep learning methods. Examining the accuracy improvement by using the proposed feedback loop strategy, requires incorporating more instances to the event classes.

Similar to any supervised learning models, we noticed some limitations that could be addressed in future work. As shown in the previous sections (such as Table 2), logbook instances contain short text (ranging from 2 to 20 tokens per instance), and utilizing recurrent deep learning algorithms such as LSTM RNNs which are heavily based on the context leads to weak performance compared to other algorithms. One possible explanation is that logbooks with short instances (sequences) are not providing sufficient context for the algorithm to make better predictions. Another could be that RNNs are notoriously difficult to train (Pascanu et al., 2013), and the LSTM models may simply require more training time to achieve similar results. There is some evidence for this, as the dataset with the most instances, which also had the second largest number of tokens per instance on average was *Faci-Main*, which is the dataset which the LSTM model had the closest performance to the CNN and BERT models, and was also the only one which the LSTM model outperformed the DNN model.

The pre-trained BERT model provided a reasonable classification performance compared to the other deep learning models, however as BERT is pre-trained on standard language, the performance when applying to logbook data was not optimal. Training or fine-tunning BERT to technical logbook data is likely to improve performance as observed in the legal and scientific domains (Chalkidis et al., 2020; Beltagy et al., 2019). As training or fine-tuning BERT requires large amounts of data, a limitation for fine-tuning a domain-specific BERT is the amount of logbook data available.

7 Conclusion and Future Work

This work focused on predictive maintenance and technical event/issue classification, with a special focus on addressing class imbalance. We acquired seven logbook datasets from three technical domains containing short instances with non-standard grammar and spelling, and many abbreviations. To address **RQ1**, we evaluated multiple strategies to address the extreme class imbalance in these datasets and we showed that the feedback loop strategy performs best, almost entirely providing the best results for the 7 different datasets and 4 different models investigated. To address **RQ2**, we empirically compared different classification algorithms (DNN, LSTM, CNN, and pre-tuned BERT). Results show that the CNN model outperforms the other classifiers. The methodology presented in this paper could be applied to other maintenance corpora from a variety of technical domains. The feedback loop approach for selecting training data is generic and could easily be applied to any learning problem with substantial class imbalances. This is useful as extreme class imbalance is a challenge at the heart of a number of natural language tasks.

In future work, we would like to fine-tune BERT using logbook data, as described in Section 6, and extend this work to datasets in other languages. The biggest challenge for these two research directions is the limited availability of logbook datasets. Furthermore, we are exploring various methods of domain adaptation and transfer learning on these datasets to further improve the performance of classification models.

Dataset	DNN	LSTM	CNN	BERT
Avi-Main	0.0020	0.0043	0.0002	0.0004
Avi-Acc	0.0011	0.0399	0.0103	0.0015
Avi-Safe	0.0000	0.0023	0.0059	0.0012
Auto-Main	0.0001	0.0181	0.0009	0.0004
Auto-Acc	0.0000	0.0055	0.0001	0.0161
Auto-Safe	0.0003	0.0106	0.0011	0.0083
Faci-Main	0.0002	0.0001	0.0003	0.0005

Table 6: Statistical significance of the various classification models between the Baseline approach and Feedback Loop approach F1 scores using the Mann-Whitney U test. Experiments indicate statistical significance with a p value of 0.05.

Acknowledgments

We would like to thanks the University of North Dakota aviation program for providing the valuable aviation maintenance logbook datasets to the MaintNet research. We further thank the aviation domain expert Zechariah Morgan for evaluating the outcomes of the various algorithms and providing valuable feedback for the aviation domain dataset. We also would like to thank the anonymous ACL reviewers for providing us with helpful comments and feedback.

References

Farhad Akhbardeh, Travis Desell, and Marcos Zampieri. 2020a. MaintNet: A collaborative open-source library for predictive maintenance language resources. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 7–11, Barcelona, Spain. International Committee on Computational Linguistics.

Farhad Akhbardeh, Travis Desell, and Marcos Zampieri. 2020b. NLP tools for predictive maintenance records in MaintNet. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 26–32, Suzhou, China. Association for Computational Linguistics.

Sadam Al-Azani and El-Sayed El-Alfy. 2017. Using word embedding and ensemble learning for highly imbalanced data sentiment analysis in short arabic text. *Procedia Computer Science*, Vol 109:359–366.

M. Altuncu, Erik Mayer, Sophia Yaliraki, and Mauricio Barahona. 2019. From free text to clusters of content in health records: an unsupervised graph partitioning approach. *Applied Network Science*, Vol 4.

Siddhartha Banerjee, Cem Akkaya, Francisco Perez-Sorrosal, and Kostas Tsioutsiouliklis. 2019. Hierarchical transfer learning for multi-label text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics ACL*, pages 6295–6300, Florence, Italy. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Connor Bowley, Marshall Mattingly, Andrew Barnas, Susan Ellis-Felege, and Travis Desell. 2019. An analysis of altitude, citizen science and a convolutional neural network feedback loop on object detection in unmanned aerial systems. *Journal of Computational Science*, Vol 34:102 – 116.

Thyago P. Carvalho, Fabrízzio A. A. M. N. Soares, Roberto Vita, Roberto da P. Francisco, João P. Basto, and Symone G. S. Alcalá. 2019. A systematic literature review of machine learning methods applied to predictive maintenance. *Computers and Industrial Engineering*, 137:106024.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Colin Cherry and Hongyu Guo. 2015. The unreasonable effectiveness of word representations for Twitter named entity recognition. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies HLT-NAACL*, pages 735–745, Denver, Colorado. Association for Computational Linguistics.

Matthias Damaschk, Tillmann Dönicke, and Florian Lux. 2019. Multiclass text classification on unbalanced, sparse and noisy data. In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 58–65, Turku, Finland. Linköping University Electronic Press.

Louise Deléger, Cyril Grouin, and Pierre Zweigenbaum. 2010. Extracting medical information from narrative patient records: The case of medication-related information. *Journal of the American Medical Informatics Association*, Vol 17:555 – 558.

Franck Dernoncourt, Ji Young Lee, and Peter Szolovits. 2017. Neural networks for joint sentence classification in medical paper abstracts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2*, pages 694–700, Valencia, Spain. Association for Computational Linguistics.

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

Kim Hammar, Shatha Jaradat, Nima Dokoohaki, and Mihhail Matskin. 2018. Deep text mining of instagram data without strong supervision. In *International Conference on Web Intelligence (WI)*, pages 158 – 165, Santiago, Chile.

Philip Kerr. 2015. Adaptive learning. *English Language Teaching (ELT) Journal*, 70(1):88–93.

Donghwa Kim, Deokseong Seo, Suhyoun Cho, and Pilsung Kang. 2019. Multi-co-training for document classification using various document representations: TF-IDF, LDA, and Doc2Vec. *Information Sciences*, Vol 477:15 – 29.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2*, pages 687–692, New

Orleans, Louisiana. Association for Computational Linguistics.

Junyi Jessy Li and Ani Nenkova. 2014. Addressing class imbalance for improved recognition of implicit discourse relations. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 142–150, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Shoushan Li, Shengfeng Ju, Guodong Zhou, and Xiaojun Li. 2012. Active learning for imbalanced sentiment classification. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 139–148, Jeju Island, Korea. Association for Computational Linguistics.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020. Dice loss for data-imbalanced NLP tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 465–476, Online. Association for Computational Linguistics.

Junyang Lin, Qi Su, Pengcheng Yang, Shuming Ma, and Xu Sun. 2018. Semantic-unit-based dilated convolution for multi-label text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4554–4564, Brussels, Belgium. Association for Computational Linguistics.

Xiaofei Ma, Peng Xu, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2019. Domain adaptation with BERT-based domain classification and data selection. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 76–83, Hong Kong, China. Association for Computational Linguistics.

Manolis Maragoudakis, Katia Kermanidis, Aristogiannis Garbis, and Nikos Fakotakis. 2006. Dealing with imbalanced data using Bayesian techniques. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

J.J. McArthur, Nima Shahbazi, Ricky Fok, Christopher Raghubar, Brandon Bortoluzzi, and Aijun An. 2018. Machine learning and bim visualization for maintenance issue classification and enhanced data collection. *Advanced Engineering Informatics*, 38:101 – 112.

Carol Midgley. 2014. *Goals, goal structures, and patterns of adaptive learning*. Routledge.

Rutu Mulkar-Mehta, Jerry Hobbs, and Eduard Hovy. 2011. Granularity in natural language discourse. In *Proceedings of the Ninth International Conference on Computational Semantics*, IWCS '11, page 360–364, USA. Association for Computational Linguistics.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted Boltzmann Machines. In *Proceedings of the 27 th International Conference on Machine Learning*, Haifa, Israel. ICML.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR.

Jon Patrick and Min Li. 2009. A cascade approach to extracting medication events. In *Proceedings of the Australasian Language Technology Association Workshop 2009*, pages 99–103, Sydney, Australia.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Alan Ritter, Mausam Mausam, Oren Etzioni, and Sam Clark. 2012. Open domain event extraction from twitter. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1104-1112:1104 – 1112.

Guergana K. Savova, James J. Masanz, Philip V. Ogren, Jiaping Zheng, Sunghwan Sohn, Karin Kipper Schuler, and Christopher G. Chute. 2010. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association : JAMIA*, 17 5:507–13.

Dinghan Shen, Martin Renqiang Min, Yitong Li, and Lawrence Carin. 2018. Learning context-sensitive convolutional filters for text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1839–1848, Brussels, Belgium. Association for Computational Linguistics.

Richard Sproat, Alan W. Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. 2001. Normalization of non-standard words. *Computer Speech & Language*, 15(3):287 – 333.

Mirac Suzgun, Yonatan Belinkov, and Stuart M. Shieber. 2019. On evaluating the generalization of LSTM models in formal languages. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 277–286.

Harish Tayyar Madabushi, Elena Kochkina, and Michael Castelle. 2019. Cost-sensitive BERT for generalisable sentence classification on imbalanced

data. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 125–134, Hong Kong, China. Association for Computational Linguistics.

Jingyun Xu, Yi Cai, Xin Wu, Xue Lei, Qingbao Huang, Ho fung Leung, and Qing Li. 2020. Incorporating context-relevant concepts into convolutional neural networks for short text classification. *Neurocomputing*, 386:42 – 53.

Meliha Yetisgen-Yildiz, Cosmin Bejan, and Mark Wurfel. 2013. Identification of patients with acute lung injury from free-text chest X-ray reports. In *Proceedings of the 2013 Workshop on Biomedical Natural Language Processing*, pages 10–17, Sofia, Bulgaria. Association for Computational Linguistics.

Zhilu Zhang and Mert R. Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 8792–8802, Red Hook, NY, USA. Curran Associates Inc.