Predicting Software Support Ticket Criticality

Using Sparse Imbalanced Data and NLP

April 14, 2020

Matt Kane
Nick Sylva
Elena Petrov

Problem Statement

Create a screening tool for incoming email support requests that predicts the severity of the issue to facilitate support ticket triage before a human can read the support email.

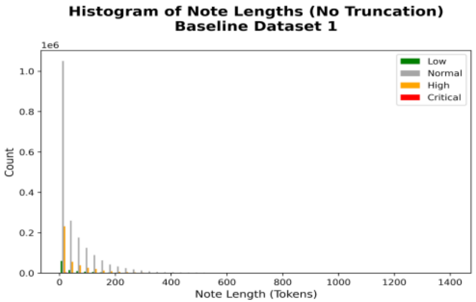
Dataset

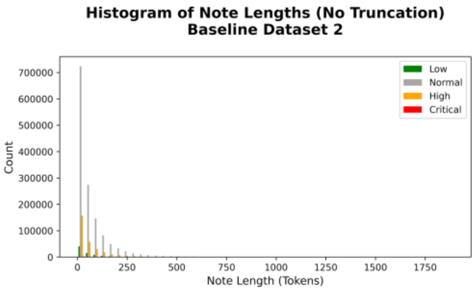
Real business dataset spanning 6,000 clients, 425,000 tickets, and ~2.7 million ticket actions over the last 10 years

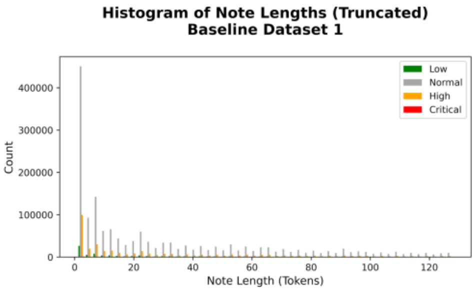
- Sourced from Greenshades, a payroll and employee services software provider for SMBs
- Raw data was ~4.3GB in CSV format

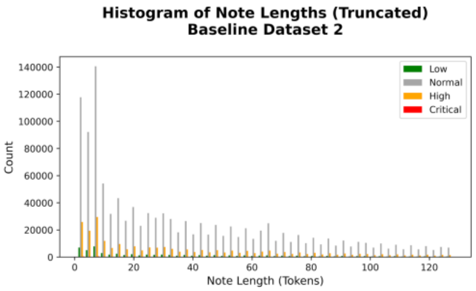
Dataset Considerations

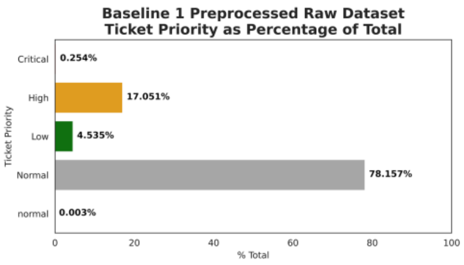
- Large dataset
- Thorough data cleansing
- Semantic augmentation
- Multi-label classification problem with highly imbalanced dataset

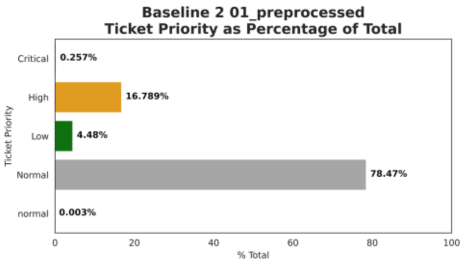












Model Specifications

Model 1: Simple Classifier with BERT Fine Tuning

- Simplest model
- BERT to single Dense layer
- Sequence representation
- Examine effectiveness of fine-tuning

Model 2: Frozen BERT and Dense NN

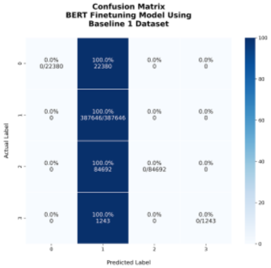
- Frozen BERT: no fine-tuning
- Sequence representation
- 4 dense layers
- Examine if dense network can pick up features better than BERT fine-tuning alone

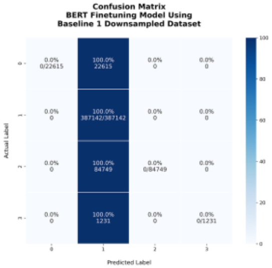
Model 3: Frozen BERT, CNN, and Dense NN

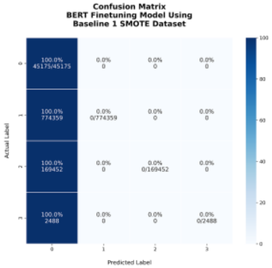
- Frozen BERT: no fine-tuning
- Full embeddings for each token
- Passed through three filter sizes of 64 filters each to pooling, concatenation, and dense network
- Pick up on relationships between token embeddings

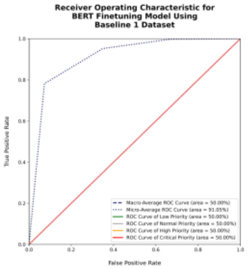
Model 1 Results: Simple Classifier with BERT Fine Tuning

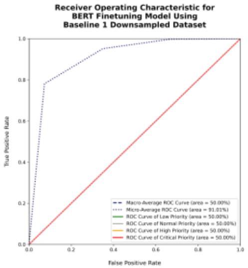
Baseline 1: Regular Normalization



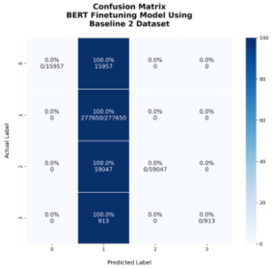


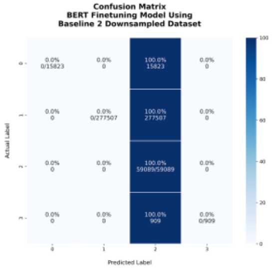


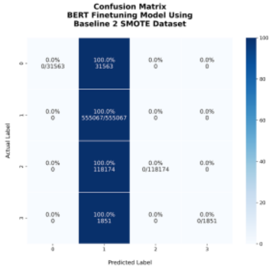


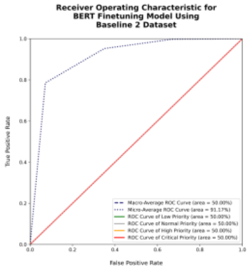


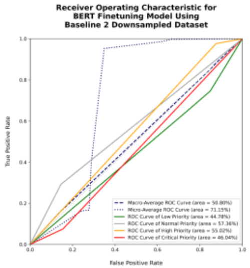
Baseline 2: Semantic Augmentation

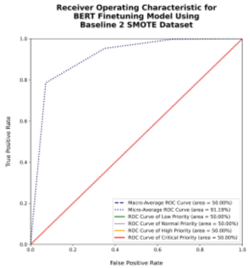






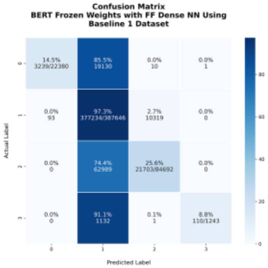


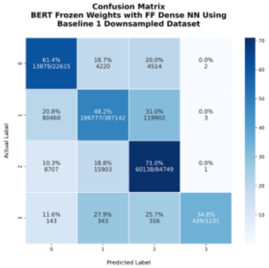


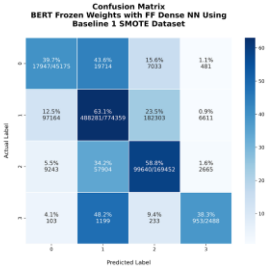


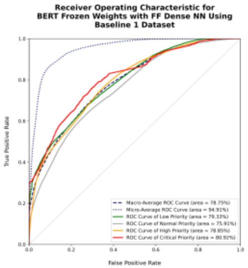
Model 2: Frozen BERT and Dense NN

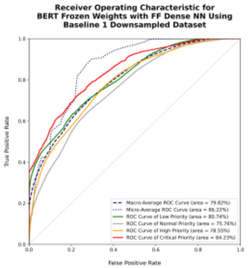
Baseline 1: Regular Normalization

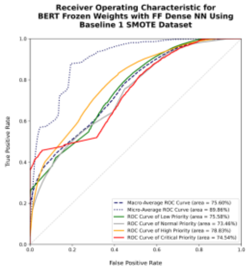




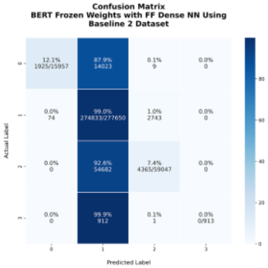


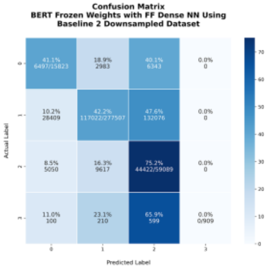


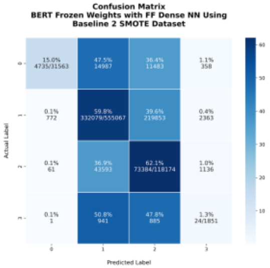


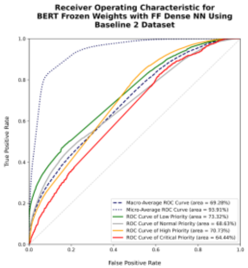


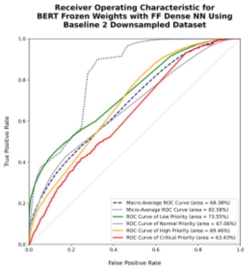
Baseline 2: Semantic Augmentation

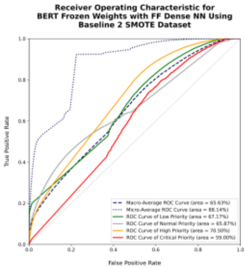






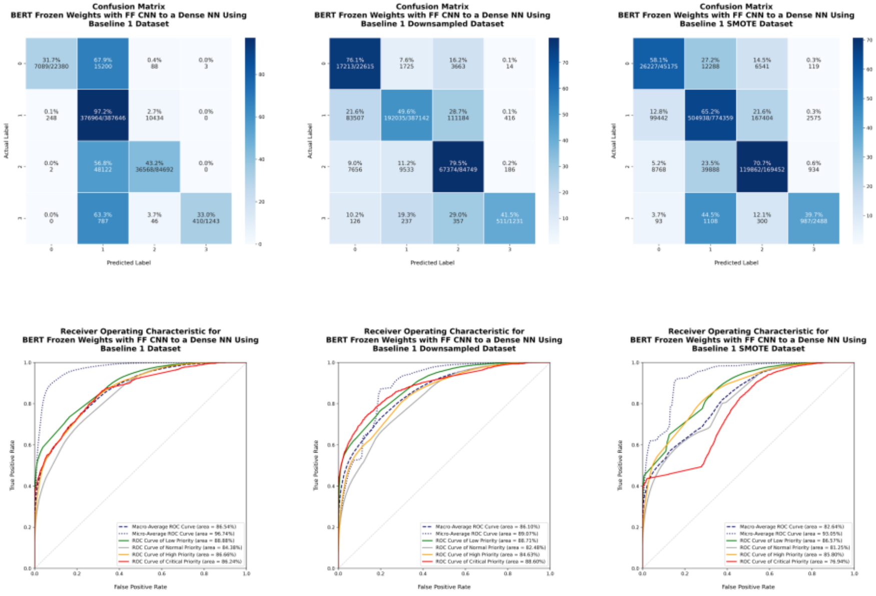




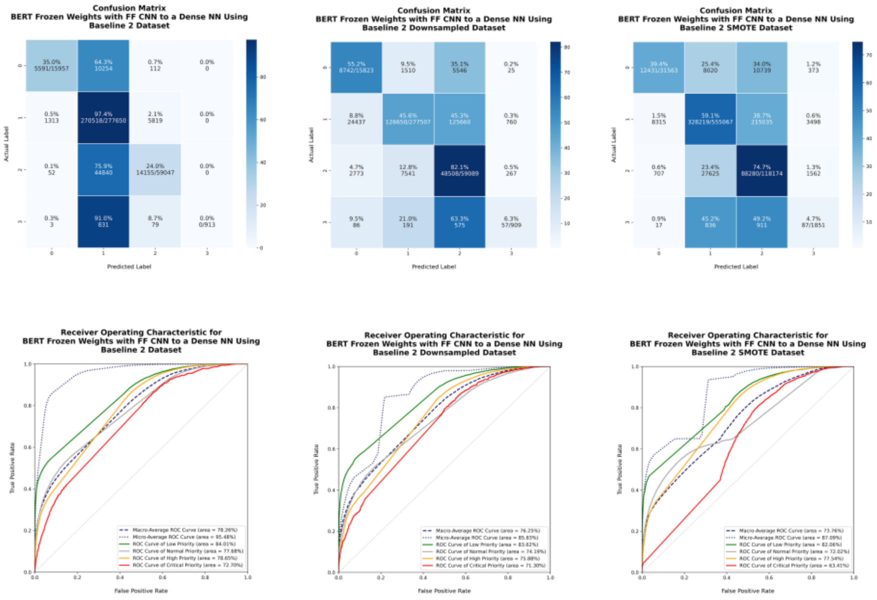


Model 3 Results: Frozen BERT, CNN, and Dense NN

Baseline 1: Regular Normalization



Baseline 2: Semantic Augmentation



Conclusions

- Fine-tuning on BERT does not solve everything
  - Bespoke dataset that is riddled with colloquialisms, abbreviations, sentence fragments and domain-specific language
- Other potential enhancements:
  - Use unfrozen BERT weights with model 3 architecture
  - Hyperparameter tuning

Results

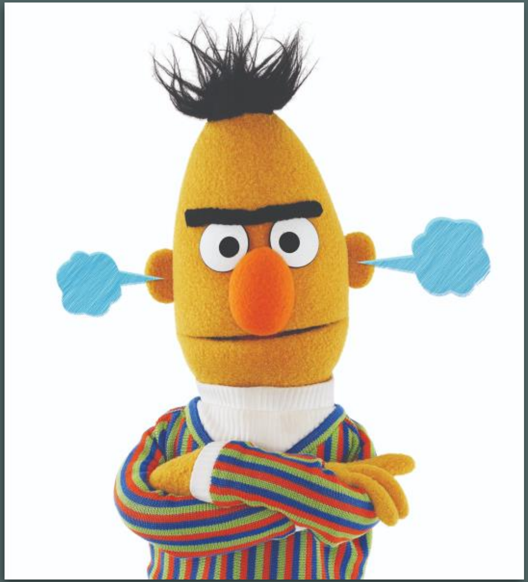
Model	Dataset	Balancing	Accuracy
Model 3	Baseline 1	None	84.89%
Model 2	Baseline 1	None	81.11%
Model 2	Baseline 2	None	79.51%
Model 1	Baseline 2	SMOTE	78.55%
Model 1	Baseline 2	None	78.53%
Model 1	Baseline 1	None	78.16%
Model 1	Baseline 1	Downsampled	78.09%
Model 3	Baseline 1	SMOTE	65.76%
Model 2	Baseline 1	SMOTE	61.20%
Model 3	Baseline 2	SMOTE	60.71%
Model 2	Baseline 2	SMOTE	58.05%
Model 3	Baseline 1	Downsampled	55.90%
Model 2	Baseline 1	Downsampled	52.69%
Model 3	Baseline 2	Downsampled	52.06%
Model 2	Baseline 2	Downsampled	47.53%
Model 1	Baseline 2	Downsampled	16.72%
Model 1	Baseline 1	SMOTE	4.56%
Model 3	Baseline 2	*** In Process ***	

Notes:

Model 1: BERT Fine Tuning Model

Model 2: BERT Frozen Weights with FF Dense NN

Model 3: BERT BERT Frozen Weights with FF CNN to a Dense NN

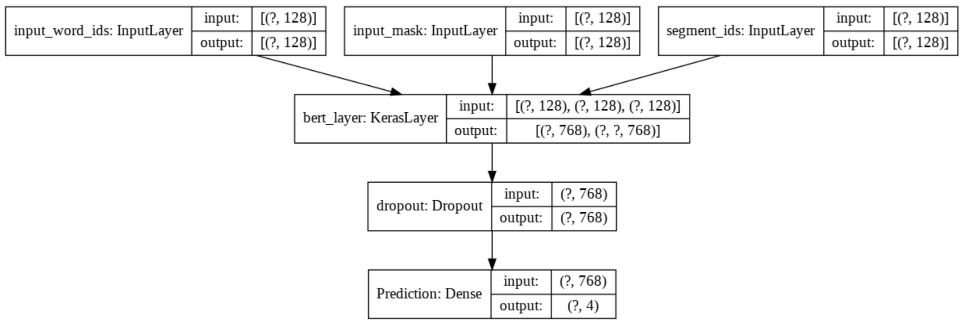


Questions

Appendices

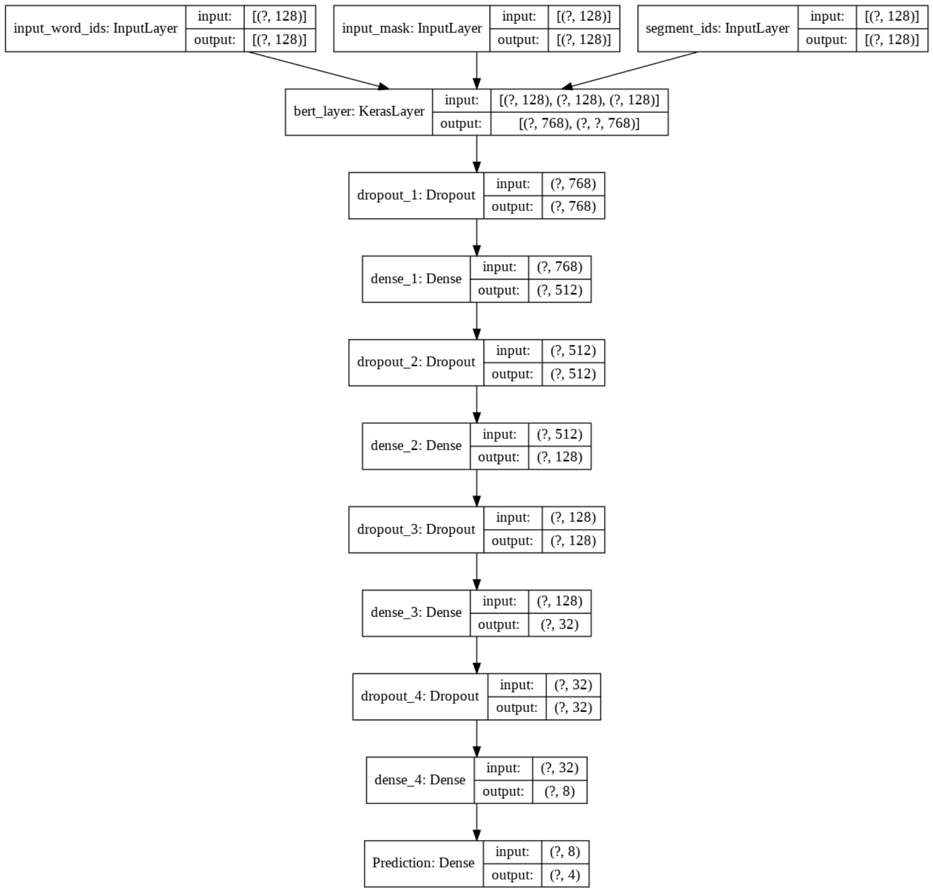
Model 1 Diagram

Simple Classifier with BERT Fine Tuning



Model 2 Diagram

Frozen BERT and Dense Neural Network



Model 3 Diagram

Frozen BERT, CNN and Feed Forward Dense Neural Network