Emergency Dispatch Triage from 911 Call Transcripts Using Multitask Transformer Classification

Arun Joshi
University of Texas at Austin
aj37332@my.utexas.edu

Abstract

Emergency dispatch calls require fast decisions from short, stressful, and incomplete conversations. This project studies whether transcript text can support three predictions: incident category, severity, and likely responder needs. The dataset combines 704 real-audio-derived 911 transcripts with 5,326 Austin Police Department (APD)-grounded synthetic transcripts, giving 6,030 documented records.

I fine-tuned `microsoft/deberta-v3-base` with three multitask designs. Model 1 used independent linear heads for category, severity, and dispatch. Model 2 used independent MLP heads with imbalance-aware losses. Model 3 used a cascade where category information fed severity, and category plus severity fed dispatch. The best overall system was Model 1, with a test composite F1 of 0.7099, category macro F1 of 0.6554, severity macro F1 of 0.6062, and dispatch positive-label macro F1 of 0.8680. Model 2 improved severity macro F1 to 0.6212, but it slightly reduced category and dispatch performance. Model 3 performed worst overall. The main conclusion is that transcript-based triage signals are learnable, but the simplest shared-encoder multitask model was the most reliable architecture in this experiment.

1. Introduction

Emergency dispatch is a high-pressure language task. A caller may be panicked, unsure of the location, or describing only part of what happened. The dispatcher still has to infer the incident type, urgency, and likely responders. Those decisions are not simple keyword matches. A phrase like "fell down" can mean a minor accident or a life-threatening injury depending on details such as breathing, consciousness, bleeding, and scene safety.

This project asks whether a modern text encoder can learn useful dispatch signals directly from 911 call transcripts. The intended use is decision support, not dispatcher replacement. Real dispatch decisions require human judgment, local policy, legal accountability, and live questioning. The goal here is narrower: test whether transcript text contains enough signal to predict operational labels.

We trained a model to perform direct transcript classification using a pretrained Transformer encoder. The model predicts three outputs: one category from 36 APD-derived classes, severity from 0 to 3, and police, EMT, and fire dispatch labels.

The project addresses three research questions:

1. Can a Transformer encoder learn dispatch-relevant predictions from 911 transcript text?
2. Does adding MLP heads and imbalance-aware losses improve over a simple independent-head multitask baseline?
3. Does a cascaded model that feeds category information into severity and dispatch improve performance?

The main finding is that a shared DeBERTa encoder can learn meaningful triage signals, but extra architecture did not improve the full system.

2. Background and Related Work

This project combines Transformer text classification, multitask learning, multilabel classification, and class-imbalance handling. Transformer models use self-attention to build contextual sequence representations (Vaswani et al., 2017). BERT showed that pretrained bidirectional encoders can be fine-tuned for many NLP tasks (Devlin et al., 2018). DeBERTa and DeBERTaV3 improve this family with disentangled attention and stronger pretraining methods (He et al., 2020; He et al., 2021), making them a good fit for dialogue where meaning depends on context.

Multitask learning lets related tasks share representations (Caruana, 1997; Ruder, 2017). Category, severity, and dispatch are related because incident type can affect urgency, and both can affect responder needs. A shared encoder is also cheaper than training three separate Transformer models.

Dispatch prediction is multilabel because police, EMT, and fire are not mutually exclusive. A crash may need police and EMT, while a fire with injuries may need all three. Multilabel learning treats each output as a binary decision while allowing co-occurrence (Zhang & Zhou, 2014). The cascade experiment is related to classifier chains, where earlier predictions become features for later ones (Read et al., 2011).

Class imbalance is also central. Emergency-call categories are long-tailed, and fire dispatch positives are rare. Macro F1 gives small classes equal weight, while precision-recall metrics are useful when positives are rare (Saito & Rehmsmeier, 2015). Reweighting methods such as class-balanced losses and focal loss are common responses (Cui et al., 2019; Lin et al., 2017).

Prior work has applied machine learning to emergency dispatch and 911 calls. Lavi et al. (2017) studied emergency medical dispatch decisions with a city partner. A 2021 Artificial Intelligence in Medicine paper used deep ensemble multitask classification for emergency medical call incidents. Nallasamy et al. (2008) studied speech recognition for minority-language emergency calls, and Burns and Moffitt (2014) used 911 transcripts for deception detection. This project is narrower: a transcript classifier for category, severity, and responder recommendation.

3. Materials and Data Sources

The project began with public 911 recordings from Kaggle. A transcription pipeline converted raw audio into text using the OpenAI audio transcription API with `gpt-4o-transcribe`, a production speech-to-text model in the same general automatic-speech-recognition family motivated by Whisper-style large-scale

weak supervision (Radford et al., 2022). The pipeline normalized recordings with ffmpeg, chunked long files when needed, and then post-processed transcripts into speaker-labeled text. After post-processing, the real-audio-derived subset contained 704 usable examples. This subset grounded the project in real emergency language, but it was too small and skewed for the full task. Severity 3 made up 68.89% of the real subset, while severity 0 had only 3 examples.

To expand the training set, the project used the Austin Police Department 911 Calls for Service 2023-2026 dataset. The APD source contained 900,827 rows with incident metadata such as type, priority, response time, final problem category, units arrived, injury counts, and disposition. APD distributions grounded synthetic transcript generation, so the generated transcripts followed real metadata and an approved taxonomy. This is related to weak supervision and synthetic data methods, where generated or noisy labels support training when direct expert labeling is limited (Ratner et al., 2017; Frénay & Verleysen, 2014; Feng et al., 2021; Wei & Zou, 2019; Sennrich et al., 2015; Bansal & Sharma, 2023).

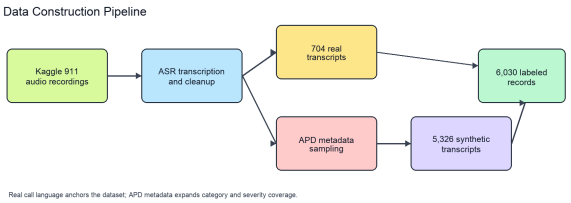
The final taxonomy removed "Other," missing labels, and extremely sparse categories such as Prostitution and Gambling. The supervised label space kept 36 categories, each with at least 100 samples.

Table 1. Dataset source summary

Source	Count	Role in project
[Kaggle 911 recordings](https://www.kaggle.com/datasets/louisteitelbaum/911-recordings)	704	Real call grounding and original transcript source
[Austin APD 911 Calls for Service](https://data.austintexas.gov/Public-Safety/APD-911-Calls-for-Service-2023-2026/e687-fx2y/about_data)	5,326	Distribution-balanced expansion using APD metadata
Final documented dataset	6,030	Combined training dataset

The final dataset columns included transcript identifiers, severity, category, police/EMT/fire dispatch labels, APD ID, and audio recording ID. The combined severity distribution was more balanced than the real-only subset: severity 0 had 827 examples, severity 1 had 1,054, severity 2 had 2,733, and severity 3 had 1,416.

Figure 1. Data construction pipeline. Raw 911 audio was transcribed with the OpenAI audio transcription API and post-processed into real transcript examples. APD metadata was used to ground synthetic transcript generation. The real and synthetic records were combined into a single transcript classification dataset.



Data Construction Pipeline

4. Problem Formulation

Each training example is one transcript. Incident category is a 36-class multiclass task. Severity is a four-class multiclass task. Dispatch is a three-label multilabel task for police, EMT, and fire.

The evaluation uses macro F1 for category and severity because both tasks have uneven class distributions. For dispatch, the evaluation reports per-label precision, recall, and F1 for the positive class, then averages positive-label F1 across police, EMT, and fire. The main composite score is:

(category_macro_f1 + severity_macro_f1 + dispatch_yes_macro_f1) / 3

This composite score is only a research metric. It weights the three project goals equally and is not a deployment utility function.

Table 2. Prediction tasks and metrics

Task	Output type	Labels	Loss	Main metric
Incident category	Multiclass	36 categories	Cross entropy	Macro F1
Severity	Multiclass	0, 1, 2, 3	Cross entropy	Macro F1
Dispatch	Multilabel binary	Police, EMT, fire	BCE with logits	Positive-label macro F1

5. Methods

5.1 Preprocessing and Split

Training records were loaded from `data_extraction/dataset.csv`, and transcript text was resolved from `data_extraction/transcripts`. Speaker tags were removed. The final notebook loaded 6,029 usable records and split them into train, validation, and test sets using a 70/15/15 category-stratified split.

Table 3. Train, validation, and test split

Split	Records	Sev 0	Sev 1	Sev 2	Sev 3
Train	4,220	592	764	1,897	967
Validation	904	130	142	412	220
Test	905	105	148	424	228

The tokenizer came from `microsoft/deberta-v3-base` with `use_fast=False`. Transcripts were padded or truncated to 300 tokens, so later information in long calls was unavailable to the classifier.

5.2 Shared Encoder

All three models fine-tune the same DeBERTaV3 base encoder. DeBERTa was chosen because the same words can imply different severity depending on negation, uncertainty, and follow-up details. The shared encoder produces one transcript representation, which is passed to task-specific heads.

5.3 Model 1: Independent Linear Heads

Model 1 is the baseline. It uses one shared DeBERTa encoder and three independent linear heads: category, severity, and dispatch. This tests whether the simplest reasonable multitask model can learn all three outputs.

5.4 Model 2: Independent MLP Heads With Imbalance Handling

Model 2 keeps independent heads but replaces each linear head with a small MLP. It also adds imbalance-aware losses: weighted cross entropy for category and severity, and positive class weights for dispatch. The dispatch weights were 1.00 for police, 1.69 for EMT, and 8.44 for fire.

5.5 Model 3: Cascaded Multitask Model

Model 3 tests an intuitive hierarchy: category should help severity, and category plus severity should help dispatch. It predicts category first, passes category information into severity, then passes category and severity information into dispatch. During training, downstream heads receive true labels. During validation and testing, they receive predicted probability-weighted embeddings, creating train-test mismatch and possible error propagation.

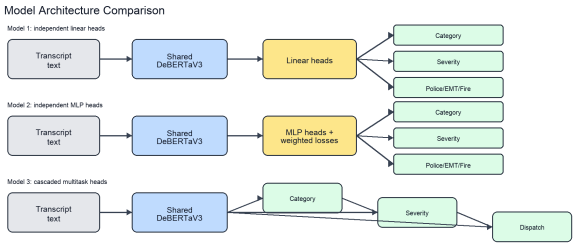
5.6 Training Setup

All models used the same training loop. The optimizer was AdamW (Kingma & Ba, 2014; Loshchilov & Hutter, 2017), with linear warmup and decay. Gradient clipping used max norm 1.0 (Pascanu et al., 2013). Early stopping used patience 4 and selected the best checkpoint by validation composite F1 (Prechelt, 1998). Category and severity label smoothing were set to 0.05 (Szegedy et al., 2015; Müller et al., 2019).

Table 4. Hyperparameter snapshot

Hyperparameter	Value
Encoder	`microsoft/deberta-v3-base`
Max token length	300
Batch size	8
Learning rate	2e-5
Epochs	12
Weight decay	0.01
Warmup ratio	0.1
Early stopping patience	4
Seed	42
MLP hidden size	256
Cascade category embedding dim	32
Cascade severity embedding dim	8
Label smoothing	0.05

Figure 2. Model architecture comparison. Model 1 uses encoder-to-linear heads. Model 2 uses encoder-to-MLP heads with imbalance-aware losses. Model 3 uses encoder-to-category, category-to-severity, and category/severity-to-dispatch cascade connections.



Model Architecture Comparison

6. Results

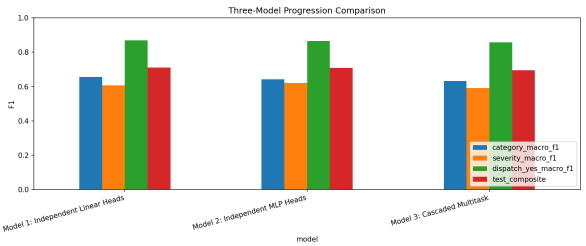
All three models learned nontrivial signal from the transcripts. Category accuracy reached roughly 69-70% across 36 classes. Severity accuracy reached roughly 64-67% across four classes. Dispatch was strongest, especially for police and EMT. The best overall result came from the simplest model.

Table 5. Headline model comparison

Model	Best Val Score	Best Epoch	Train Time Min	Inference Sec	Category Macro F1	Severity Macro F1	Dispatch Yes Macro F1	Test Composite
Model 1: Independent Linear Heads	0.716276	11	21.479856	9.176597	0.655376	0.606157	0.868038	0.709857
Model 2: Independent MLP Heads	0.709368	12	22.028191	9.361078	0.641436	0.621166	0.864151	0.708918
Model 3: Cascaded Multitask	0.692366	12	22.150645	9.277790	0.632761	0.591097	0.857911	0.693923

Model 1 had the best validation composite, test composite, category macro F1, dispatch positive-label macro F1, and inference time. Model 2 improved severity but lost ground on category and dispatch. Model 3 had the weakest headline performance despite encoding the clearest hierarchy.

Figure 3. Model progression comparison. The progression chart summarizes category macro F1, severity macro F1, dispatch positive-label macro F1, and composite F1 across the three model architectures.



Model Progression Comparison

Table 6. Delta relative to Model 1

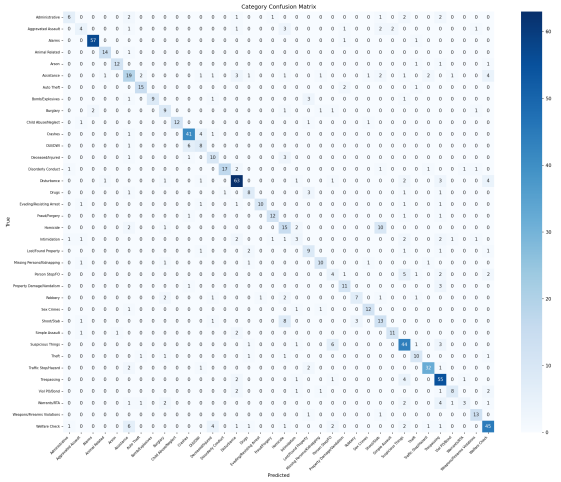
Model	Category Macro F1	Severity Macro F1	Dispatch Yes Macro F1	Test Composite
Model 2 vs Model 1	-0.0139	+0.0150	-0.0039	-0.0009
Model 3 vs Model 1	-0.0226	-0.0151	-0.0101	-0.0159

6.1 Category Results

Model 1 achieved category accuracy of 0.6972, category macro F1 of 0.6554, and weighted F1 of 0.6940. The confusion matrix had a clear diagonal, so the model was separating incident types rather than guessing. Strong classes included Alarms, Animal Related, Child Abuse/Neglect, Arson, Crashes, Traffic Stop/Hazard, Fraud/Forgery, Auto Theft, and Weapons/Firearms Violations.

The weakest classes were more ambiguous or sparse. Model 1 F1 was 0.24 for Intimidation, 0.27 for Person Stop/FO, 0.30 for Warrants/RTA, and 0.32 for Aggravated Assault. Many errors happened among overlapping call types such as assistance, welfare checks, injuries, assaults, suspicious events, and person-stop calls. This is understandable because callers may describe similar scenes in language that maps to different official labels.

Figure 4. Category confusion matrix for Model 1. The diagonal structure shows that the model learned meaningful category separation, while off-diagonal errors cluster among overlapping incident types.



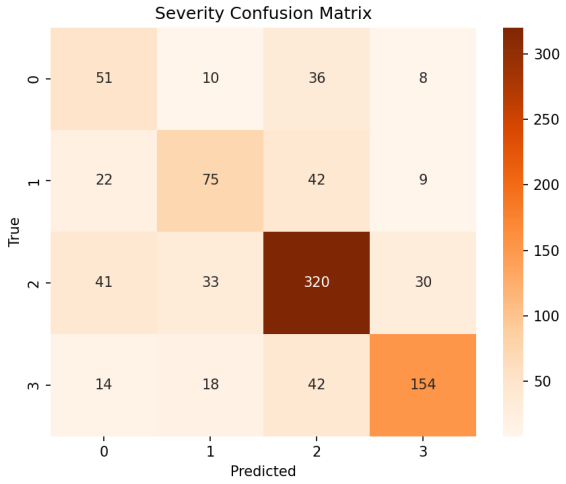
Model 1 Category Confusion Matrix

6.2 Severity Results

Severity was where Model 2 helped most. Model 1 severity macro F1 was 0.6062. Model 2 improved it to 0.6212, while Model 3 fell to 0.5911. The improvement mainly came from severity 0: Model 2 raised severity-0 F1 from 0.44 to 0.51 and improved recall from 51/105 to 67/105 test examples.

That gain had a cost. Model 2 moved more severity-2 examples into other classes. The class weighting made the model more willing to predict rare low-severity cases, which improved recall but redistributed errors. It helped severity alone, but not the overall composite score.

Figure 5. Severity confusion matrix for Model 1. The matrix shows that severity 2 and 3 were easier for the model, while severity 0 and 1 were more often confused with severity 2.



Model 1 Severity Confusion Matrix

6.3 Dispatch Results

Dispatch was the strongest task across all models. Model 1 positive-label F1 was 0.9822 for police, 0.8785 for EMT, and 0.7435 for fire. Model 2 had 0.9783 for police, 0.8793 for EMT, and 0.7349 for fire. Model 3 had 0.9783 for police, 0.8845 for EMT, and 0.7109 for fire.

Table 7. Dispatch positive-label F1 by model

Model	Police F1	EMT F1	Fire F1	Positive-label macro F1
Model 1	0.9822	0.8785	0.7435	0.8680
Model 2	0.9783	0.8793	0.7349	0.8642
Model 3	0.9783	0.8845	0.7109	0.8579

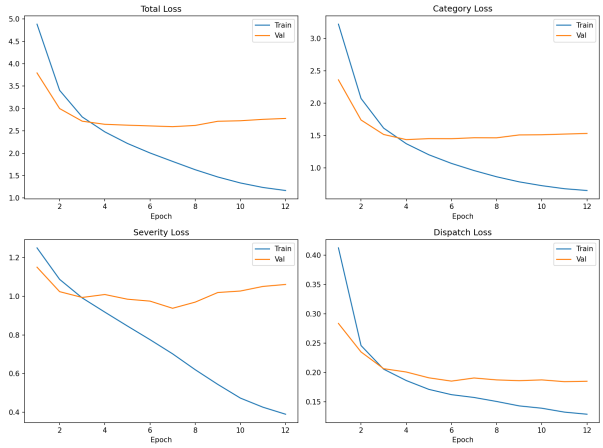
Police was easiest because police positives dominate the dataset. EMT was also strong. Fire was hardest because positives are rare and fire dispatch can be subtle. Model 2's positive weighting increased fire recall from 0.7172 to 0.7980, but precision dropped from 0.7717 to 0.6810, so fire F1 did not improve.

6.4 Training Behavior

All three models showed some overfitting. Training loss kept falling while validation loss flattened and later rose. Training accuracy also continued to improve after validation performance plateaued. Best-checkpoint selection by validation composite F1 therefore mattered. Model 1 peaked at epoch 11 with validation

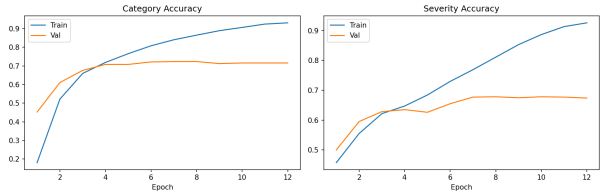
composite 0.7163. Models 2 and 3 peaked at epoch 12, but their validation scores stayed below Model 1.

Figure 6. Model 1 training and validation loss curves. The training loss continued downward while validation loss flattened and later rose, which is consistent with overfitting after the early epochs.



Model 1 Loss Curves

Figure 7. Model 1 training and validation accuracy curves. Training accuracy kept improving after validation accuracy plateaued, which supports the use of validation-based checkpoint selection.



Model 1 Accuracy Curves

Runtime differences were small. Training took about 21.5 to 22.2 minutes per model, and test-set inference took about 9.2 to 9.4 seconds.

7. Discussion

The first research question asked whether a Transformer encoder can learn dispatch-relevant predictions from transcript text. Within this dataset, the answer is yes. The best test composite F1 was 0.7099. Category accuracy reached about 69.7% across 36 classes, severity accuracy reached 66.3% for the best severity model, and dispatch positive-label macro F1 reached 0.8680.

The second question asked whether MLP heads and imbalance-aware losses improve the baseline. The answer is partial. Model 2 improved severity macro F1 from 0.6062 to 0.6212 and improved recall for severity 0. However, category macro F1 dropped by 0.0139, dispatch positive-label macro F1 dropped by 0.0039, and the composite score dropped slightly. Model 2 did not fail; it changed the tradeoff. It would be more attractive if low-severity recall were the main priority.

The third question asked whether a cascade helps. It did not in this experiment. Model 3 was lower than Model 1 on every headline test metric, with a composite score of 0.6939. Category, severity, and dispatch are clearly related, but the implemented cascade introduced train-test mismatch. During training, downstream heads saw true category and severity labels. During evaluation, they saw model predictions. When upstream predictions were uncertain or wrong, downstream outputs inherited that noise.

The main lesson is that simple baselines matter. The independent linear-head model had less structure, but it avoided forcing uncertain intermediate predictions into later decisions. Model 2 showed that weighting can help a weak severity class. Model 3 showed that an intuitive hierarchy is not automatically useful.

8. Limitations and Threats to Validity

These results support the modeling approach, not real dispatch deployment. The largest limitation is that the dataset is mostly synthetic: 704 real-audio-derived examples and 5,326 APD-grounded synthetic examples. APD grounding helps preserve realistic category and priority patterns, but generated transcripts may be cleaner and more label-aligned than real calls.

The APD source is also police-centered. It strongly grounds police incident categories, but EMT and fire labels are less directly observed from a unified dispatch system. This matters because police prediction was strongest while fire remained hardest.

Some category labels were also vague or overlapping. The same transcript could reasonably fall into multiple categories, such as Assistance, Welfare Check, Disturbance, Suspicious Things, or Deceased/Injured, depending on what officers found after arriving. In real dispatch work, the final category and severity can change after police visit the incident and collect more information. The APD data is post-incident administrative data, while the model only sees the initial transcript. That creates cases where there may be no clear way for the model to learn the final label from the call text alone.

Severity and dispatch labels also represent a project labeling policy, not pure observed ground truth from dispatchers. A real system would need local protocol, resource availability, caller safety, and live questioning. The category distribution remains uneven as well, with several labels near the 100-sample floor.

The final evaluation is transcript-only. It does not measure end-to-end audio-to-dispatch error propagation from noisy audio, overlapping speech, accents, panic, dropped calls, or transcription errors. It also uses a fixed 300-token input length, a fixed 0.5 dispatch threshold, and no external validation set.

9. Conclusion

This project tested whether 911 call transcripts can support machine learning predictions for incident category, severity, and responder dispatch. The answer is yes, with important limits. A DeBERTa-based multitask model learned useful signal across all three tasks, and the best model reached a test composite F1 of 0.7099.

The strongest architecture was Model 1: one shared DeBERTa encoder with independent linear heads. Model 2 improved severity macro F1, especially for low-severity calls, but did not improve the full multitask score. Model 3 showed that explicitly cascading category into severity and dispatch did not help, likely because train-test mismatch and upstream error propagation outweighed the benefit of modeling task dependency.

Future work should focus on more real transcripts, directly observed dispatch labels, external validation, calibrated dispatch

thresholds, longer-context modeling, and human-in-the-loop review. The final takeaway is that transcript-based emergency triage is learnable, but the simplest multitask design was the most reliable model here.

References

Bansal, P., & Sharma, A. (2023). Large language models as annotators: Enhancing generalization of NLP models at minimal cost. arXiv. https://arxiv.org/abs/2306.15766

Burns, M. B., & Moffitt, K. C. (2014). Automated deception detection of 911 call transcripts. *Security Informatics, 3*, 8. https://doi.org/10.1186/s13388-014-0008-2

Caruana, R. (1997). Multitask learning. *Machine Learning, 28*, 41-75. https://doi.org/10.1023/A:1007379606734

City of Austin. (2026). APD 911 Calls for Service 2023-2026. Austin Open Data. https://data.austintexas.gov/Public-Safety/APD-911-Calls-for-Service-2023-2026/e687-fx2y/about_data

Cui, Y., Jia, M., Lin, T.-Y., Song, Y., & Belongie, S. (2019). Class-balanced loss based on effective number of samples. arXiv. https://arxiv.org/abs/1901.05555

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv. https://arxiv.org/abs/1810.04805

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., & Hovy, E. (2021). A survey of data augmentation approaches for NLP. *Findings of ACL 2021*. https://aclanthology.org/2021.findings-acl.84/

Frénay, B., & Verleysen, M. (2014). Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems, 25*(5), 845-869. https://doi.org/10.1109/TNNLS.2013.2292894

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. arXiv. https://arxiv.org/abs/1706.04599

He, P., Gao, J., & Chen, W. (2021). DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. arXiv. https://arxiv.org/abs/2111.09543

He, P., Liu, X., Gao, J., & Chen, W. (2020). DeBERTa: Decoding-enhanced BERT with disentangled attention. arXiv. https://arxiv.org/abs/2006.03654

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv. https://arxiv.org/abs/1412.6980

Kaggle. (n.d.). 911 recordings. https://www.kaggle.com/datasets/louisteitelbaum/911-recordings

Lavi, D., et al. (2017). Using machine learning to improve emergency medical dispatch decisions. *KDD 2017*. https://www.kdd.org/kdd2017/papers/view/using-machine-learning-to-improve-emergency-medical-dispatch-decisions

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2017). Focal loss for dense object detection. arXiv. https://arxiv.org/abs/1708.02002

Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. arXiv. https://arxiv.org/abs/1711.05101

Müller, R., Kornblith, S., & Hinton, G. (2019). When does label smoothing help? arXiv. https://arxiv.org/abs/1906.02629

Nallasamy, U., Black, A. W., Schultz, T., & Frederking, R. (2008). NineOneOne: Recognizing and classifying speech for handling minority language emergency calls. *LREC 2008*. https://www.cs.cmu.edu/~awb/papers/lrec2008/LREC2008-911-CMU.pdf

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. arXiv. https://arxiv.org/abs/1211.5063

Prechelt, L. (1998). Early stopping - but when? In *Neural Networks: Tricks of the Trade*. https://page.mi.fu-berlin.de/prechelt/Biblio/stop_tricks1997.pdf

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust speech recognition via large-scale weak supervision. arXiv. https://arxiv.org/abs/2212.04356

Ratner, A., Bach, S. H., Ehrenberg, H., Fries, J., Wu, S., & Ré, C. (2017). Snorkel: Rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment, 11*(3), 269-282. https://vldb.org/pvldb/vol11/p269-ratner.pdf

Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2011). Classifier chains for multi-label classification. *Machine Learning, 85*, 333-359. https://doi.org/10.1007/s10994-011-5256-5

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. arXiv. https://arxiv.org/abs/1706.05098

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE, 10*(3), e0118432. https://doi.org/10.1371/journal.pone.0118432

Sennrich, R., Haddow, B., & Birch, A. (2015). Improving neural machine translation models with monolingual data. arXiv. https://arxiv.org/abs/1511.06709

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). Rethinking the Inception architecture for computer vision. arXiv. https://arxiv.org/abs/1512.00567

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. arXiv. https://arxiv.org/abs/1706.03762

Wei, J., & Zou, K. (2019). EDA: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv. https://arxiv.org/abs/1901.11196

Zhang, M.-L., & Zhou, Z.-H. (2014). A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering, 26*(8), 1819-1837. https://www.lamda.nju.edu.cn/publication/tkde13rev.pdf

Deep ensemble multitask classification of emergency medical call incidents combining multimodal data improves emergency medical dispatch. (2021). *Artificial Intelligence in Medicine, 117*, 102088. https://doi.org/10.1016/j.artmed.2021.102088

Appendix A. AI Use Disclosure

AI-based tools were used in this project for limited implementation support, language refinement, transcript processing, and synthetic data generation as described in the data construction methodology. Specifically, OpenAI audio transcription with `gpt-4o-transcribe` and synthetic transcript generation formed part of the dataset preparation pipeline.

All model design decisions, experimental setup, evaluation procedures, interpretation of results, and final conclusions were developed, reviewed, and finalized by the student. Large language

models were additionally used as auxiliary programming and editing support tools during development but were not used to generate the core research contributions or claims of this work.