A methodology for emergency calls severity prediction: from pre-processing to BERT-based classifiers

Marianne Abi Kanaan[1,2][0000−0002−0362−2863],
Jean-François Couchot[1][0000−0001−6437−5598],
Christophe Guyeux[1][0000−0003−0195−4378], David Laiymani[1][0000−0003−2580−6660],
Talar Atechian[2][0000−0002−9140−1402], and Rony Darazi[2][0000−0001−6057−6245]

[1] FEMTO-ST Institute, CNRS, Université de Franche-Comté, Besançon, France
{marianne.abi_kanaan,jean-francois.couchot,
christophe.guyeux,david.laiymani}@univ-fcomte.fr
[2] TICKET Lab, Université Antonine (UA), Baabda, Lebanon
{marianne.abikanaan,talar.atechian,rony.darazi}@ua.edu.lb

Abstract. Emergency call centers are often required to properly assess and prioritise emergency situations pre-intervention, in order to provide the required assistance to the callers efficiently. In this paper, we present an end-to-end pipeline for emergency calls analysis. Such a tool can be found useful as it is possible for the intervention team to misinterpret the severity of the situation or mis-prioritise callers. The data used throughout this work is one week's worth of emergency call recordings provided by the French SDIS 25 firemen station, located in the Doubs. We pre-process the calls and evaluate several artificial intelligence models in the classification of callers' situation as either severe or non-severe. We demonstrate through our results that it is possible, with the right selection of algorithms, to predict if the call will result in a serious injury with a 71% accuracy, based on the caller's speech only. This shows that it is indeed possible to assist emergency centers with an autonomous tool that is capable of analysing the caller's description of their situation and assigning an appropriate priority to their call.

Keywords: Emergency Calls · Text Classification · Transformers · Speech-To-Text · Machine Learning · Audio Processing.

1 Introduction

Speech is the main form of communication in human conversations. The analysis of speech can provide many insights that characterize a conversation's intent, nature, emotions... and many more indicators. As such, a speech analysis system can prove to be useful in various contexts.

One such domain is emergency call centers, as they can sometimes face an overload of calls, which in turn leads to mis-prioritisation of the cases. Given that early medical interventions can lead to less fatalities in the cases of severe injuries, it seems that the assistance of operators in their job is essential. This is the case of French Emergency centers for example. In France, just as 911 operators handle emergency situations in the U.S., this work is handled by the SDIS (Service Départemental d'Incendie et de Secours) department of a specific region. They are call centers operated by firemen, and they handle emergency situations in that area 24/7. Following emergency calls, some situations often require the intervention of an emergency team, and as such, there is always a risk of misinterpreting the needs of the intervention i.e. number and nature of resources needed, etc.

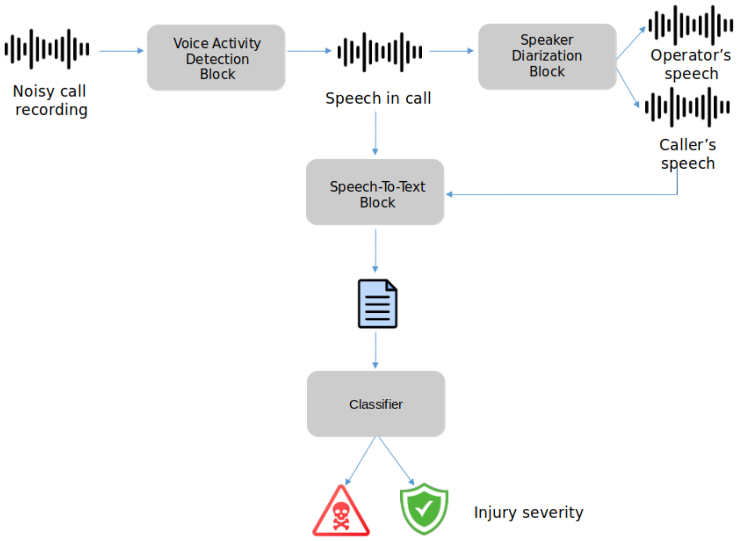

Fig. 1. Emergency phone calls processing pipeline.

Therefore, in this work, our goal is to implement a speech analysis system for emergency centers, and eventually answer the following question: Is it possible to manage incidents more efficiently through a system that can process a phone call, and thereby provide a relevant assessment of the emergency at-hand ?

This article describes our attempt to develop a prototype that aims at assisting operators in handling emergency situations by automatically analysing incoming phone calls and assessing the severity of the caller's situation. The emergency calls used in this work are equivalent to one week's worth of real-life calls provided by the French SDIS 25 firemen station, located in the Doubs region of France.

The starting point of our pipeline is a "voice activity detection" block. This allows us to extract intervals that contain speech activity in the audio streams and discard segments where speech is absent. The next step is the application of speaker diarization on each audio file to extract the caller's speech into a separate signal. Two speech-to-text systems are then evaluated in the automatic transcription of the audio files into text. And finally, we implement an analysis block, which consists of a classifier that labels the transcribed text as a "high severity" case or a "low-severity" case. Our experiments show that it is possible to predict the severity of an emergency call with a 71% accuracy based on the caller's speech only. This result is highly influenced by the choice of classifier and speech-to-text system.

The remainder of the paper is structured as follows: Section 2 covers related works that tackle emergency calls categorisation. The proposed methodology for labeling the calls is described in detail in Section 3. The experiments and their evaluation are reported in Section 4. Finally, a conclusion, some limitations of the work and possible future directions are summarized in Section 5.

2 Related Work

Various works have attempted to use speech analysis in healthcare applications, specifically in diagnosing callers to an emergency department with a specific condition. The work in [8] uses a machine learning framework developed by a Danish company to predict cardiac arrests based on automatic transcriptions of a call. The framework achieved a higher sensitivity compared to the medical dispatcher (84.1% vs 72.5%). In [6], the authors describe their study with multiple machine learning classifiers in the classification of manually annotated emergency calls transcriptions into a pre-hospital diagnosis. They do not use the text as is, but rather extract descriptors such as TF-IDF embeddings [33], and train several machine learning algorithms (SVM, Linear Regression...) using these feature vectors. Their most accurate model, an SVM using TF-IDF, achieves a 95% accuracy on unseen data. In our work, we attempt to provide a more generalised analysis of the situation at-hand and assess its risk regardless of the diagnosis. The work in [30] attempts to automate the prioritisation of 911 calls using SVM algorithms with written transcriptions of the calls provided by a security service, using techniques such as lemmatization and pruning. Their model labels the call as either high-priority or low-priority. Its best result is a recall rate of 86%, a precision rate of 75%, and an f1-score of 80%. In our case, considering that our aim is to predict the severity of the call as quickly as possible before the intervention of the firemen's team, using deep learning methods would provide a better performance as they can act as feature extractors without the need for an additional pre-processing step [22].

With the emergence of transformer-based language models, most works have shifted their focus on models such as GPT [35] and BERT [36], which require less text pre-processing, and are available in pre-trained versions on various language-understanding work. One such use of transformers is [17], where the authors attempt to associate a diagnosis to each transcription of an emergency call in a French medical emergency department. The authors pre-train the generative model GPT-2 in an unsupervised manner on a subset of the dataset, then re-train this model on the classification of another part of the annotations. The used dataset is a collection of reports made by intervening physicians, medical assistants, and paramedics. Their f1-scores on each class range from 47.9% to 80%. Similarly, [37] use a pre-trained Chinese BERT to automatically categorise emergency reports with a 91.55% weighted f1-score. The model is trained with a custom loss function in an attempt to overcome the data imbalance issue in their dataset. In our work, given that we are in a binary classification scenario, the data imbalance issue can be resolved by randomly and manually removing samples from the dominating class, which leaves us with a well-balanced dataset.

What sets our work aside from the previously described papers is, first, to the best of our knowledge, no work has attempted to develop an end-to-end pipeline to assess emergency calls' severity in a center. Furthermore, our aim is to detect potentially severe calls, regardless of the diagnosis of the caller. A fall for example can at times seriously injure an individual, while at other times lead to minor or no injuries. This aspect of our data increases the difficulty of our task. We hope that our text analysis component will be able to pick up on specific cues, terms, or patterns in the caller's speech that can go undetected by the operator, in order to predict the possible outcome of the situation. As such, we attempt to perform classification of automatically annotated call recordings, which poses an additional challenge compared to the previously described works, as the transcribed text can contain many errors. To implement the analysis block of our pipeline, transformer-based models seem to be more suitable in our case, as they require less text pre-processing, which is ideal for a future real-time implementation of the system. In addition, we lean towards pre-trained BERT models as opposed to GPT models as in [17], since GPT is originally a generative model [35], and requires an additional step of unsupervised learning on our dataset.

3 Methods

3.1 Dataset

As previously mentioned, the emergency calls used in this work were provided by the SDIS 25, an emergency department in the Doubs region in France. The dataset consists of one week's worth of data, i.e. 904 audio recordings of phone calls in the French language in WMA format. This dataset was constructed by filtering out some of the calls provided by the SDIS: calls between operators, calls between operators and policemen, calls between medical professionals and dispatchers... These conversations are irrelevant to our task since they often discuss the details of a specific intervention on site, whereas our goal is to evaluate the needs of the intervention before the team goes through with it, and based only on the analysis of a non-professional's speech.

Some statistics regarding the duration and number of words of the calls on each version of the dataset post-processing (Callers-Only/Callers-Operators, see next section), are reported in Table 1. A typical conversation consists of the operator interrogating the caller to gather information and provide help to the victim. The recordings are accompanied by a file that includes the reason of the call (e.g car accident, loss of consciousness...) and the state of the victim after the intervention, which consists of three possibilities: lightly injured, severely injured, or deceased. Since "deceased" is a minority class, it was grouped with the "severely injured" category. As such, a "lightly injured" label describes minor injuries such as scratches, small fractures, small wounds... and so on, whereas a "severely injured" label can mean the victim is either deceased or severely wounded. It should be noted that the level of injury does not always depend on the diagnosis in some cases. If, for example, a victim was drowning and the team was able to intervene early, the victim would likely have no injuries. Similarly, a minor fire in a caller's kitchen could grow and result in major injuries in case the team arrives late on site. This is one imperfection in our dataset that makes the prediction of the severity more challenging.

A confidentiality agreement was signed with the SDIS that restricts us from sharing the dataset and the text classification model, since the latter could leak callers' data if shared with a third-party.

In this work, we are interested in labeling a call as either "low-severity", which consists of the "lightly injured" cases, or "high-severity" which includes the "severely injured" calls. In the cases where the same call is related to several victims (in the case of a car accident for example), where each victim has a different level of injury, we include the call once with the most severe injury as the label. We balance out the dataset by manually removing some examples that are labeled "lightly injured", since these types of injuries are more common than the severe ones. The resulting dataset is therefore made up of 49.78% "low-severity" calls, and of 50.22% "high-severity" calls.

Table 1. Statistics about the datasets used in this work.

		Min	Max	Average
Callers-Only	Call length (seconds)	4.25	410.77	136.84
	Number of words in call	76	8257	2502
Callers-Operators	Call length (seconds)	8.5	540.91	207.88
	Number of words in call	144	8985	3814

3.2 Audio pre-processing

Speech Detection and Speaker Diarization The emergency calls recordings in question contain many parts that are irrelevant to our task and could introduce additional noise. Such parts include the answering machine and the waiting music sounds, or the parts where the caller is waiting for someone to answer their call and there isn't any voice activity. For this reason, we apply Voice Activity Detection (VAD) on these recordings to extract the segments that contain voices. VAD allows the detection of speech regions in a given audio recording. Many studies [27, 5, 10] have shown that the application of voice activity detection in speech-analysis systems can produce cleaner data and achieve a higher performance. We implement this using Pyannote.audio, an open-source collection of neural building blocks for speaker diarization [9]. In addition, since the aim of this work is to predict if a caller is seriously injured or not through the analysis of their speech, our work requires an additional step of extracting the caller's speech into a separate signal. As such, we created a Speaker Diarization block with the use of Pyannote.audio [9].

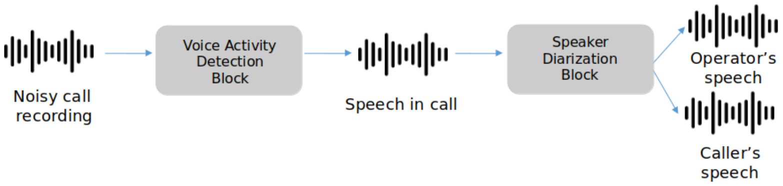

Fig. 2. Emergency call recordings pre-processing phase.

The reason for choosing the aforementioned solution is that it is open-source and can be used offline, meaning that the data will remain protected and uncompromised. In addition, Pyannote.audio provides a pre-trained model for speaker diarization, which eliminates the need to train a model from scratch. It also provides the lowest recorded diarization error rate (DER) in the literature, when tested on French "ETAPE" corpus [9]. Once we obtained the segments for each speaker, we manually separate the segments of the caller and the operator. We aim to automate this process for the real-world application of our method by training a model that could automatically recognize the operator's speech.

We finally re-join the segments of each speaker to obtain one complete audio recording for each speaker. The audio pre-processing phase is illustrated in Figure 2.

As illustrated in Figure 3, we later train several classifiers separately on two versions of the dataset: one with the callers-operators dialogues, and one with only the caller's speech. We then evaluate the trained models and compare the results obtained on each version of the dataset.

Speech To Text For the Automatic Speech Recognition (ASR) component, we compare two speech-to-text systems: Whisper [34] and Vosk API [2]. Whisper is a simple encoder-decoder Transformer [34], trained on 680,000 hours of diverse multilingual data collected from the web. VOSK API is a speech recognition toolkit based on Kaldi [2]. It offers various language-specific models, in both large and lightweight versions of the models. The first reason for choosing both of these systems is that they are open source and offline, which ensures that the privacy of our dataset remains protected. A second reason is the proven efficiency of both of these systems on the French language. Whisper has achieved low word error rates (WER) on several French datasets [34], as the highest WER for the large version is 14.7%. Vosk API has been equally successfully used in French speech transcription applications [15], achieving decent word error rates compared to Google Cloud's Speech-To-Text [1]. We select the large version of the multilingual Whisper and the French Vosk.

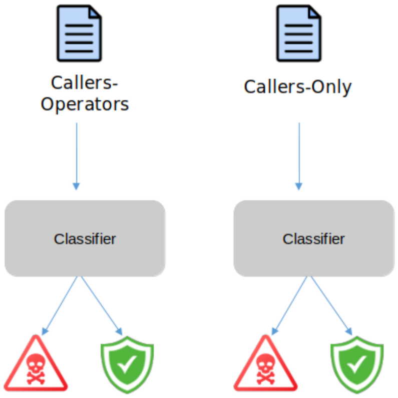

Fig. 3. Training of classifiers on callers-operators speech vs. callers-only speech.

3.3 Emergency Call Severity Prediction

In this section, we describe the implementation of machine learning and deep learning algorithms to predict the severity of a call, equivalent to the level of injury of the victim. On the one hand, we attempt to analyse the audio calls as they are, by training machine learning algorithms on their acoustic features. On the other hand, we use NLP methods to analyse the transcriptions of these audio calls. These methods range from simple models such as LSTMs and CNNs, to more advanced models such as transformers. We then compare the results obtained using each approach.

Audio Classification For this implementation, we fragment each audio file into 10 seconds long fragments, as in several speech classification works [25, 21]. The fragments are overlapped by 5 seconds in order to minimise the loss of context in the speech post-fragmentation. We then extract a set of acoustic feature vectors for each audio fragment using Librosa [28] at a sample rate of 8000 Hz. We choose to extract 40 Mel Frequency Cepstral Coefficients (MFCCs) for each fragment. MFCCs are frequently used to represent speech [24, 7], as they can represent sound as it is heard by the human ear.

We train our machine learning model in a speaker-independent manner, meaning we completely separate all fragments and avoid fragments leaking from speakers that are included in the test set. As for our audio classifier, and given the fact that our dataset is of a relatively small size for complex audio applications, we opt for machine learning algorithms instead of deep neural networks, as they can achieve decent results on limited data. We train and evaluate an XGBoost model [11] on the MFCC features, since it has been proven that they can achieve competitive results in several audio classification tasks in a clinical context [24, 18].

Transcriptions Classification Most NLP applications nowadays have moved from using RNN-based models, such as LSTMs and GRUs, to using transformers, a type of neural network that utilizes self-attention to learn context in text [29]. CamemBERT [26] is a pre-trained French transformer, based on the RoBERTa architecture (robustly optimized BERT pretraining approach) [23], a variant of BERT [36]. The BERT models [36] are multipurpose pre-trained models that can be trained on several NLP tasks such as text classification, named entity recognition, and many more tasks.

Several steps are required to fine-tune CamemBERT on our dataset. The text is first tokenized with the uncased CamemBERT tokenizer. All transcriptions are either truncated or padded to match a maximum length that we set based on the results of a hyperparameters search (described in the next section). Finally, we use attention masks to allow the model to differentiate between padded and real tokens. We use the base CamemBERT model, and fine-tune it with a single linear classification layer.

In order to obtain a better idea of the difficulty of our text classification task, we establish additional baselines that can be compared to our BERT-based approach. A first baseline is a simple LSTM network, that consists of an embedding input layer, followed by three LSTM layers of size 256, and a 30% dropout layer to reduce the effect of overfitting. The second baseline is an optimized version of the well-known TextCNN model [19]. Some works, such as [16], have demonstrated that CNNs or Hierarchical Neural Networks, can sometimes achieve better results in clinical text classification tasks, compared to BERT. For this reason, we implement a state-of-the-art version of the TextCNN architecture [38].The network consists of three Convolutional 2D layers, each with 512 filters of sizes 2,3,5 respectively. Once convolution is applied on the text matrix, it is followed by a 1-max pooling layer, which extracts the largest number from each feature map. The resulting feature vectors are concatenated into one, and followed by a final layer with sigmoid activation function to output one of the two labels. We use the GloVe multilingual 300 dimensional embedding [32] to represent our vocabulary in both the LSTM and TextCNN. Our final test is a multi-lingual approach using the XLM-RoBERTa model in its base version [12]. XLM-RoBERTa was pre-trained on a massive corpus from 100 languages, making it a strong candidate for use in multilingual applications and in the support of cross-lingual language processing tasks.

4 Experiments and Evaluation

4.1 Experiments and Hyperparameter Tuning

The training computations are completed using the PyTorch [31] and XGBoost [3] frameworks on an NVIDIA Tesla V100 GPU with 32 GB of memory. Across all our experimentations, we split the dataset using the 80/10/10 method: 80% training data, 10% for validation, and 10% for testing, and report the mean accuracy of 10-fold cross-validation runs. First, we perform a hyperparameters selection for our deep learning models through the Grid Search approach, using Optuna library [4]. For the optimisation of CamemBERT, we base our search on the range of values recommended by BERT's authors [14]. The obtained hyperparameters for all models are reported in Table 2. Developed models are not necessarily the same depending on whether the operators' speech is integrated or not. Consequently, their parameters also change. The operators line of this table indicates if, in this model, the operator's speech is present ("w" for with), absent ("wo" for without). In the case where the same hyperparameters have been chosen, independently of the presence of the operator's speech, w/wo (for with or without) is shown.

Table 2. Optimized hyperparameters of the classifiers obtained using a Grid Search.

Hyperparameters	CamemBERT		XLM-RoBERTa	TextCNN	LSTM	XGBoost	
Operators	w	wo	w/wo	w/wo	w/wo	w	wo
Sequence Length	384	512	512	512	512	-	-
Learning Rate	3e-5	5e-5	5e-5	6e-4	1e-4	1e-4	1e-3
Epsilon	1e-7	1e-5	1e-5	-	-	-	-
Decay	-	-	-	1e-6	1e-6	-	-
Batch Size	16	8	8	8	8	-	-
Estimators	-	-	-	-	-	1e4	1e4
Max Depth	-	-	-	-	-	9	9

Given that the maximum supported sequence length in CamemBERT is 512, we process our text to match this length for the training of the TextCNN and the LSTM, in order to allow all models to learn from the same context. This leads to the discarding of words beyond the 512th word, since the average sequence length ranges from 2502 to 3812 (see section 3.1). Even though some informative parts are lost, we don't consider this a limitation, since our end goal is to assist emergency center operators before the end of the call, with a minimal amount of speech content. We train CamemBERT on each run for 15 epochs instead of the recommended number of 4, since we found that training the network for longer led to higher accuracies on the test set. For each run, we evaluate the model on the test set after each epoch, and select the model with the highest accuracy among all 15 epochs. The same procedure is applied to XLM-RoBERTa.

As mentioned in Sect. 3.2, we separately train our models twice: first on the callers-operators transcriptions, and then on the callers-only transcriptions. Finally for the training of the LSTM and TextCNN, we use early stopping to interrupt training when the validation loss stops decreasing, and checkpoint the model with the lowest validation loss. We use the Adam optimizer [20] as all the networks' optimiser.

4.2 Results

We report in Table 3 the mean accuracy with a 95% confidence interval of the 10-fold cross validation runs for each model. The scores are reported for each combination of data type, speech-to-text system, model, and version of the dataset.

The results show that among the tested models, the CamemBERT one that was trained on the complete caller-operator transcriptions provided the highest accuracy. It achieved a slightly better result compared to the callers-only CamemBERT. In fact, CamemBERT was able to provide approximately the same performance with or without the operator's part of the conversation. This shows that in a scenario where the emergency center is trying to automatically prioritise a call, the caller's description of their situation would be enough for the system to assign them a priority and assess their situation. The audio XGBoost models obtained the lowest scores, ranging from 49.56 to 50.5%, close to the accuracy of random binary guesses. This proves that in an emergency context, acoustic features in a call recording on their own are not informative enough of a caller's situation.

Table 3. Classification accuracies for the models with a 95% confidence interval.

Data Type	Speech-To-Text system	Model	Callers-Only Accuracy	Callers-Operator Accuracy
Audio (MFCCs)	-	XGBoost	49.56 ± 2.25%	50.5 ± 0.90%
Text	Whisper	LSTM	57.83 ± 2.93%	58.22 ± 5.49%
	Whisper	TextCNN	57.56 ± 6.21%	63.96 ± 4.01%
	Whisper	XLM-RoBERTa	55.55 ± 3.2%	56.0 ± 2.14%
	Vosk API	CamemBERT	68 ± 4.29%	69.55 ± 4.64%
	Whisper	CamemBERT	**71.2 ± 3.02%**	**72.3 ± 2.66%**

As for the baseline LSTM and TextCNN trained on the GloVe embeddings, they both underperfom compared to CamemBERT. They achieve similar accuracies on the callers-only dataset, whereas the TextCNN performs better on the callers-operators dataset. This demonstrates the robustness of BERT-based models in text classification tasks, as unlike the LSTM and TextCNN, CamemBERT was able to achieve decent results on automatically transcribed noisy textual data. Additionally, the Whisper-transcribed text has surprisingly led to higher scores compared to the Vosk API transcriptions, knowing that Whisper is a multilingual model, while the Vosk model was specifically fine-tuned on French language data. This indicates that the multilingual Whisper is a suitable choice for a speech-to-text component in a call center. We can also note that the combination of Whisper/CamemBERT achieves the most stable results, since it has a lower margin of error (confidence interval) compared to other text models. Finally, note that the multi-lingual XLM-RoBERTa delivers sub-par results and seems to have difficulties to generalize good performances with a small french dataset.

Table 4 represents the confusion matrices for the CamemBERT model for callers-only dataset and the complete callers-operators data. The matrices show that it is easier for both models to predict the "Low-Severity" cases than the "High-Severity" ones. Surprisingly, the callers-only model tends to predict severe cases slightly more accurately, whereas the callers-operators model performs better on the non-severe cases. We plan to evaluate this further with a larger dataset in a future work.

Table 4. Confusion matrix of CamemBERT with 71% and 72% accuracy respectively on each testing set.

		High-Severity	Low-Severity
Callers-Only	High-Severity	TP=27	FN=18
	Low-Severity	FP=8	TN=38
Callers-Operators	High-Severity	TP=26	FN=19
	Low-Severity	FP=6	TN=40

5 Conclusion

In the present work, it was concluded that with the appropriate system design choices, it is possible to predict the severity of an emergency call based on only the caller's description of their situation. It is worth noting that the reported severity is the one resulting from the intervention of the emergency team, and may not always conform to the diagnosis of the caller. This, alongside the absence of accurately annotated calls, poses an additional challenge in this work.

The results in this study imply that the feasibility of such an application depends on an adequate analysis of the call's transcriptions through a BERT-based model, preferably specific to the language of the dataset, CamemBERT [26] for instance in the case of this work. The choice of the speech-to-text system also highly influences the accuracy of the predictions, as unlike the classifier, a multilingual model such as Whisper [34] is robust enough to transcribe phone calls with a higher accuracy compared to other language-specific systems [2].

One of the main limitations of this study is that the performance of the system was evaluated based on the accuracy of its predictions, whereas an interesting additional evaluation would be one concerning its computational efficiency, in terms of speed and resources consumption. As such, we plan on implementing several improvements to the system in the future. On the one hand, we aim to improve our system's predictions accuracy by augmenting CamemBERT with the emotional features of the phone calls obtained through speech emotion recognition models. We also aim to attempt the treatment of longer sequences of text, using models such as Longformers [13]. On the other hand, we will evaluate the system's performance in terms of efficiency and inference speed, as we plan to obtain an analysis of an emergency call as the conversation is going. To do so, we will evaluate and optimise each of the pipeline's components performance separately (VAD, Speech-To-Text, CamemBERT).

Acknowledgements We would like to thank the SDIS 25 (Service Départemental d'Incendie et de Secours du Doubs) for their invaluable assistance, for sharing their needs and ideas, and for trusting us with the dataset used in this work. Computations have been performed on the supercomputer facilities of the "Mésocentre de calcul de Franche-Comté". This work is (partially) supported by the EIPHI Graduate School (contract ANR-17-EURE-0002). It is also supported by the Safar scholarship program, co-funded by the French embassy in Lebanon, and the Agence Universitaire de la Francophonie AUF-BMO, in the framework of the Interuniversity Scientific Cooperation Project WeBel.

References

1. Google cloud speech to text. https://cloud.google.com/speech-to-text, accessed: 2022-09-30
2. Vosk offline speech recognition api. https://alphacephei.com/vosk/, accessed: 2022-09-30
3. Xgboost extreme gradient boosting. https://github.com/dmlc/xgboost, accessed: 2022-11-01
4. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 2623–2631 (2019)
5. Alghifari, M.F., Gunawan, T.S., Qadri, S.A.A., Kartiwi, M., Janin, Z., et al.: On the use of voice activity detection in speech emotion recognition. Bulletin of Electrical Engineering and Informatics **8**(4), 1324–1332 (2019)
6. Anthony, T., Mishra, A.K., Stassen, W., Son, J.: The feasibility of using machine learning to classify calls to south african emergency dispatch centres according to prehospital diagnosis, by utilising caller descriptions of the incident. In: Healthcare. vol. 9, p. 1107. MDPI (2021)
7. Bhavan, A., Chauhan, P., Shah, R.R., et al.: Bagged support vector machines for emotion recognition from speech. Knowledge-Based Systems **184**, 104886 (2019)

8. Blomberg, S.N., Folke, F., Ersbøll, A.K., Christensen, H.C., Torp-Pedersen, C., Sayre, M.R., Counts, C.R., Lippert, F.K.: Machine learning as a supportive tool to recognize cardiac arrest in emergency calls. Resuscitation **138**, 322–329 (2019)
9. Bredin, H., Yin, R., Coria, J.M., Gelly, G., Korshunov, P., Lavechin, M., Fustes, D., Titeux, H., Bouaziz, W., Gill, M.P.: pyannote.audio: neural building blocks for speaker diarization. In: ICASSP 2020, IEEE International Conference on Acoustics, Speech, and Signal Processing (2020)
10. Cen, L., Wu, F., Yu, Z.L., Hu, F.: A real-time speech emotion recognition system and its application in online learning. In: Emotions, technology, design, and learning, pp. 27–46. Elsevier (2016)
11. Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., et al.: Xgboost: extreme gradient boosting. R package version 0.4-2 **1**(4), 1–4 (2015)
12. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. CoRR **abs/1911.02116** (2019), http://arxiv.org/abs/1911.02116
13. Dai, X., Chalkidis, I., Darkner, S., Elliott, D.: Revisiting transformer-based models for long document classification. arXiv preprint arXiv:2204.06683 (2022)
14. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
15. Fadel, W., Araf, I., Bouchentouf, T., Buvet, P.A., Bourzeix, F., Bourja, O.: Which french speech recognition system for assistant robots? In: 2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET). pp. 1–5. IEEE (2022)
16. Gao, S., Alawad, M., Young, M.T., Gounley, J., Schaefferkoetter, N., Yoon, H.J., Wu, X.C., Durbin, E.B., Doherty, J., Stroup, A., et al.: Limitations of transformers on clinical text classification. IEEE journal of biomedical and health informatics **25**(9), 3596–3607 (2021)
17. Gil-Jardiné, C., Chenais, G., Pradeau, C., Tentillier, E., Revel, P., Combes, X., Galinski, M., Tellier, E., Lagarde, E.: Trends in reasons for emergency calls during the covid-19 crisis in the department of gironde, france using artificial neural network for natural language classification. Scandinavian journal of trauma, resuscitation and emergency medicine **29**(1), 1–9 (2021)
18. Irawati, M.E., Zakaria, H.: Classification model for covid-19 detection through recording of cough using xgboost classifier algorithm. In: 2021 International Symposium on Electronics and Smart Devices (ISESD). pp. 1–5. IEEE (2021)
19. Kim, Y.: Convolutional neural networks for sentence classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (08 2014). https://doi.org/10.3115/v1/D14-1181
20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
21. Kong, Q., Cao, Y., Iqbal, T., Wang, Y., Wang, W., Plumbley, M.D.: Panns: Large-scale pretrained audio neural networks for audio pattern recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing **28**, 2880–2894 (2020)
22. Liang, H., Sun, X., Sun, Y., Gao, Y.: Text feature extraction based on deep learning: a review. EURASIP journal on wireless communications and networking **2017**(1), 1–12 (2017)

23. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
24. Long, J.M., Yan, Z.F., Shen, Y.L., Liu, W.J., Wei, Q.Y.: Detection of epilepsy using mfcc-based feature and xgboost. In: 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI). pp. 1–4. IEEE (2018)
25. Luz, S., Haider, F., de la Fuente, S., Fromm, D., MacWhinney, B.: Alzheimer's dementia recognition through spontaneous speech: The adress challenge. arXiv preprint arXiv:2004.06833 (2020)
26. Martin, L., Muller, B., Suárez, P.J.O., Dupont, Y., Romary, L., de La Clergerie, É.V., Seddah, D., Sagot, B.: Camembert: a tasty french language model. arXiv preprint arXiv:1911.03894 (2019)
27. McDuff, D., Rowan, K., Choudhury, P., Wolk, J., Pham, T., Czerwinski, M.: A multimodal emotion sensing platform for building emotion-aware applications. arXiv preprint arXiv:1903.12133 (2019)
28. McFee, B., Raffel, C., Liang, D., Ellis, D.P., McVicar, M., Battenberg, E., Nieto, O.: librosa: Audio and music signal analysis in python. In: Proceedings of the 14th python in science conference. vol. 8, pp. 18–25 (2015)
29. Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., Gao, J.: Deep learning–based text classification: a comprehensive review. ACM computing surveys (CSUR) **54**(3), 1–40 (2021)
30. Orellana, M., Trujillo, A., Acosta, M.I.: A methodology to predict emergency call high-priority: Case study ecu-911. In: 2020 Seventh International Conference on eDemocracy & eGovernment (ICEDEG). pp. 243–247. IEEE (2020)
31. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
32. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
33. Qaiser, S., Ali, R.: Text mining: use of tf-idf to examine the relevance of words to documents. International Journal of Computer Applications **181**(1), 25–29 (2018)
34. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision. arXiv preprint arXiv:2212.04356 (2022)
35. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)
36. Tenney, I., Das, D., Pavlick, E.: Bert rediscovers the classical nlp pipeline. arXiv preprint arXiv:1905.05950 (2019)
37. Wang, Z., Wang, L., Huang, C., Luo, X.: Bert-based chinese text classification for emergency domain with a novel loss function. arXiv preprint arXiv:2104.04197 (2021)
38. Zhang, Y., Wallace, B.: A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820 (2015)