Real-Time Speech Emotion and Sentiment Recognition for Interactive Dialogue Systems

Dario Bertero, Farhad Bin Siddique, Chien-Sheng Wu,
Yan Wan, Ricky Ho Yin Chan and **Pascale Fung**
Human Language Technology Center
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
[dbertero, fsiddique]@connect.ust.hk, b01901045@ntu.edu.tw,
ywanad@connect.ust.hk, eehychan@ust.hk, pascale@ece.ust.hk

Abstract

In this paper, we describe our approach of enabling an interactive dialogue system to recognize user emotion and sentiment in real-time. These modules allow otherwise conventional dialogue systems to have "empathy" and answer to the user while being aware of their emotion and intent. Emotion recognition from speech previously consists of feature engineering and machine learning where the first stage causes delay in decoding time. We describe a CNN model to extract emotion from raw speech input without feature engineering. This approach even achieves an impressive average of 65.7% accuracy on six emotion categories, a 4.5% improvement when compared to the conventional feature based SVM classification. A separate, CNN-based sentiment analysis module recognizes sentiments from speech recognition results, with 82.5 F-measure on human-machine dialogues when trained with out-of-domain data.

1 Introduction

Interactive dialogue systems and chatbots have been around for a while. Some, though not all, systems have statistical and machine learning modules to enable them to improve overtime. With the pervasiveness of such systems on mobile devices, expectations of user experience have also increased. We expect human-machine dialogues to get closer to human-human dialogues. One important factor is that we expect machines to understand our emotions and intent and respond with empathy.

We propose a module of emotion and sentiment recognition for an interactive dialogue system. This module enables the system to assess the user's current emotional state and sentiment, and thereby decide the appropriate response at every dialogue state. The dialogue management system handles the mixed-initiative dialogues while taking into account user emotion and sentiment, in addition to query content. Emotion and sentiment recognition enables our system to handle user queries previously unseen in training data. Positive user queries containing positive emotion and sentiment label would have a positive response, and similarly a negatively labeled statement would have a negative response. Examples are shown below:

User: *I lost my job.*
Response: *Sorry to hear that. Success is in never giving up.*
User: *I just graduated from college!*
Response: *Congratulations! I am happy for you.*
User: *I went on a vacation last month and it was pretty bad, I lost all my luggage*
Response: *That doesn't sound so good. Hope your next vacation will be a good one.*
User: *My last vacation was amazing, I loved it!*
Response: *That sounds great. I would like to travel with you.*

Meanwhile, dialogue systems like this need to have real-time recognition of user emotion and sentiment. Previous approaches of emotion recognition from speech involve feature engineering (Schuller et al., 2009; Schuller et al., 2010) as a first step which invariably causes delay in decoding. So we are interested in investigating a method to avoid feature engineering and instead use a Convolutional Neural

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1042–1047,
Austin, Texas, November 1-5, 2016. ©2016 Association for Computational Linguistics

Network to extract emotion from raw audio input directly.

2 Speech Recognition

Our acoustic data is obtained from various public domain corpora and LDC corpora, comprised of 1385hrs of speech. We use Kaldi speech recognition toolkit (Povey et al., 2011) to train our acoustic models. We train deep neural network hidden Markov models (DNN-HMMs) using the raw audio together with encode-decode parallel audio. We apply layer-wise training of restricted Boltzmann machines (RBM) (Hinton, 2010), frame cross-entropy training with mini-batch stochastic gradient descent (SGD) and sequence discriminative training using state Minimum Bayes Risk (sMBR) criterion.

The text data, of approximately 90 million sentences, includes acoustic training transcriptions, filtered sentences of Google 1 billion word LM benchmark (Chelba et al., 2013), and other multiple domains (web news, music, weather). Our decoder allows streaming of raw audio or CELP encoded data through TCP/IP or HTTP protocol, and performs decoding in real time. The ASR system achieves 7.6% word error rate on our clean speech test data[1].

3 Real-Time Emotion Recognition from Time-Domain Raw Audio Input

In recent years, we have seen successful systems that gave high classification accuracies on benchmark datasets of emotional speech (Mairesse et al., 2007) or music genres and moods (Schermerhorn and Scheutz, 2011).

Most of such work consists of two main steps, namely feature extraction and classifier learning, which is tedious and time-consuming. Extracting high and low level features (Schuller et al., 2009), and computing over windows of audio signals typically takes a few dozen seconds to do for each utterance, making the response time less than real-time instantaneous, which users have come to expect from interactive systems. It also requires a lot of hand tuning. In order to bypass feature engineering, the current direction is to explore methods that can recognize emotion or mood directly from time-domain audio signals. One approach that has shown great potential is using Convolutional Neural Networks. In the following sections, we compare an approach of using CNN without feature engineering to a method that uses audio features with a SVM classifier.

3.1 Dataset

For our experiments on emotion recognition with raw audio, we built a dataset from the TED-LIUM corpus release 2 (Rousseau et al., 2014). It includes 207 hours of speech extracted from 1495 TED talks. We annotated the data with an existing commercial API followed by manual correction. We use these 6 categories: criticism, anxiety, anger, loneliness, happiness, and sadness. We obtained a total of 2389 segments for the criticism category, 3855 for anxiety, 12708 for anger, 3618 for loneliness, 8070 for happy and 1824 for sadness. The segments have an average length slightly above 13 seconds.

3.2 Convolutional Neural Network model

The Convolutional Neural Network (CNN) model using raw audio as input is shown in Figure 1. The raw audio samples are first down-sampled at 8 kHz, in order to optimize between the sampling rate and representation memory efficiency in case of longer segments. The CNN is designed with a single filter for real-time processing. We set a convolution window of size 200, which corresponds to 25 ms, and an overlapping step size of 50, equal to around 6 ms. The convolution layer performs the feature extraction, and models the variations among neighboring, overlapping frames. The subsequent max-pooling combines the contributions of all the frames, and gives as output a segment-based vector. This is then fed into a fully connected layer before the final softmax layer. These last layers perform a similar function as those of a fully connected Deep Neural Network (DNN), mapping the max-pooling output into a probabilistic distribution over the desired emotional output categories.

During decoding the processing time increases linearly with the length of the audio input segment. Thus the largest time contribution is due to the computations inside the network (He and Sun, 2015), which with a single convolution layer can be performed in negligible time for single utterances.

[1] https://catalog.ldc.upenn.edu/LDC94S13A

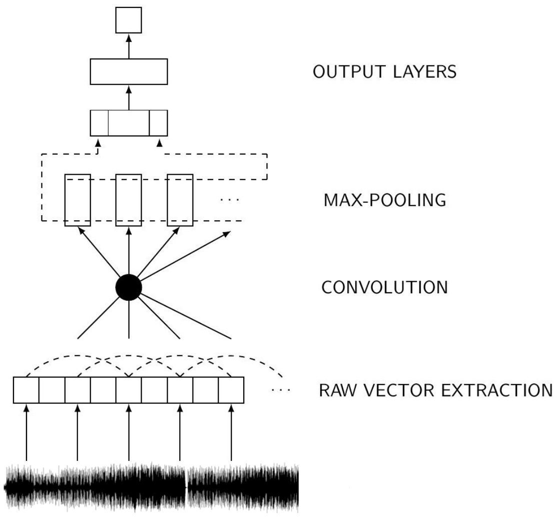

Figure 1: Convolutional Neural Network model for emotion classification from raw audio samples.

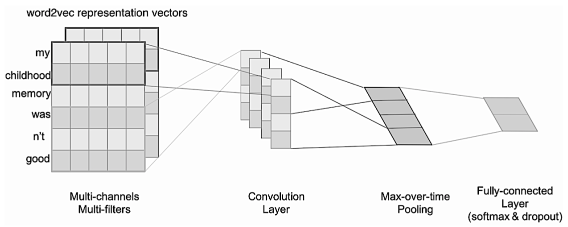

Figure 2: Convolutional neural network model for sentiment classification

4 Sentiment Inference from Speech and Text

Convolutional Neural Networks (CNNs) have recently achieved remarkably strong performance also on the practically important task of sentence classification (Johnson and Zhang, 2014; Kalchbrenner et al., 2014; Kim, 2014). In our approach, we use a CNN-based classifier with Word2Vec to analyze the sentiment of recognized speech.

We train a CNN with one layer of convolution and max pooling (Collobert et al., 2011) on top of word embedding vectors trained on the Google News corpus (Mikolov et al., 2013) of size 300. We apply on top of the word vectors a convolutional sliding window of size 3, 4 and 5 to represent multiple features. We then apply a max-pooling operation over the output vectors of the convolutional layer, that allows the model to pick up the most valuable information wherever it happens in the input sentence, and give as output a fixed-length sentence encoding vector.

Emotion class	SVM	CNN
Criticism/Cynicism	55.0	61.2
Defensiveness/Anxiety	56.3	62.0
Hostility/Anger	72.8	72.9
Loneliness/Unfulfillment	61.1	66.6
Love/Happiness	50.9	60.1
Sadness/Sorrow	71.1	71.4
Average	61.2	**65.7**

Table 1: Accuracy obtained, percentage, in the Convolutional Neural Network model for emotion classification from raw audio samples.

We employ two distinct CNN channels: the first uses word embedding vectors directly as input, while the second fine-tunes them via back propagation (Kim, 2014). All the hidden layer dimensions are set to 100. The final softmax layer takes as input the concatenated sentence encoding vectors of the two channels, and gives as output is the probability distribution over a binary classification for sentiment analysis of text transcribed from speech by our speech recognizer.

To improve the performance of sentiment classification in real time conversation, we compare the performance on the Movie Review dataset used in Kim (2014) with the Twitter sentiment 140[2] dataset. This twitter dataset contains a total of 1.6M sentences with positive and negative sentiment labels. Before training the CNN model we apply some preprocessing as mentioned in Go et al. (2009).

5 Experiments

5.1 Experimental setup

For the speech emotion detection module we setup our experiments as binary classification tasks, in which each segment is classified as either part of a particular emotion category or not. For each category the negative samples were chosen randomly from the clips that did not belong to the positive category. We took 80% of the data as training set, and 10% each as development and test set. The development set was used to tune the hyperparameters and determine the early stopping condition. We implemented our CNN with the THEANO framework

[2] `www.sentiment140.com`

Corpus	Average Length	Size	Vocabulary Size	Words in Word2vec
Movie Review	20	10662	18765	16448
Twitter	12.97	1600000	273761	79663

Table 2: Corpus statistics for text sentiment experiments with CNN.

Model	Accuracy	Precision	Recall	F-score
CNN model (trained on Movie Review dataset)	67.8%	91.2%	63.5%	74.8
LIWC (keyword based)	**73.5%**	**80.3%**	77.3%	77.7
CNN model (trained on Twitter dataset)	72.17%	78.64%	**86.69%**	**82.5**

Table 3: Sentiment analysis result on human-machine dialogue when trained from Twitter and Movie Review dataset

(Bergstra et al., 2010). We chose rectified linear as the non-linear function for the hidden layers, as it generally provided better performance over other functions. We used standard backpropagation training, with momentum set to 0.9 and initial learning rate to 10^{-5}. As a baseline we used a linear-kernel SVM model from the LibSVM (Chang and Lin, 2011) library with the INTERSPEECH 2009 emotion feature set (Schuller et al., 2009), extracted with openSMILE (Eyben et al., 2010). These features are computed from a series of input frames and output a single static summary vector, e.g, the smooth methods, maximum and minimum value, mean value of the features from the frames (Liscombe et al., 2003).

A similar one-layer CNN setup was used also for the sentiment module, again with rectified linear as the activation function. As our dataset contains many neutral samples, we trained two distinct CNNs: one for positive sentiment and one for negative, and showed the average results among the two categories. For each of the two training corpora we took 10% as development set. We used as baseline a method that uses positive and emotion keywords from the Linguistic Inquiry and Word Count (LIWC 2015) dictionary (Pennebaker et al., 2015).

5.2 Results and discussion

5.2.1 Speech emotion recognition

Results obtained by this module are shown in Table 1. In all the emotion classes considered our CNN model outperformed the SVM baseline, sometimes marginally (in the angry and sad classes), sometimes more significantly (happy and criticism classes). It is particularly important to point out that our CNN does not use any kind of preprocessed features. The lower results for some categories, even on the SVM baseline, may be a sign of inaccuracy in manual labeling. We plan to work to improve both the dataset, with hand-labeled samples, and periodically retrain the model as ongoing work.

Processing time is another key factor of our system. We ran an evaluation of the time needed to perform all the operations required by our system (down-sampling, audio samples extraction and classification) on a commercial laptop. The system we used is a Lenovo x250 laptop with a Intel i5 CPU, 8 Gb RAM, an SSD hard disk and running Linux Ubuntu 16.04. Our classifier took an average of 162 ms over 10 segments randomly chosen from our corpus of length greater than 13 s, which corresponds to 13 ms per second of speech, hence achieving real-time performance on typical utterances. The key of the low processing time is the lightweight structure of the CNN, which uses only one filter. We replicated the evaluations with the same 10 segments on a two-filter CNN, where the second filter spans over 250 ms windows. Although we obtained higher performance with this structure in our preliminary experiments, the processing time raised to 6.067 s, which corresponds to around 500 ms per second of speech. This is over one order of magnitude higher than the one filter configuration, making it less suitable for time constrained applications such as dialogue systems.

5.2.2 Sentiment inference from ASR

Results obtained by this module are shown in Table 3. Our CNN model got a 6.1% relative improvement on F-score over the baseline when trained with the larger Twitter dataset. The keyword based method got a slightly better accuracy and precision and a much lower recall on our relatively small human-machine dialogue dataset (821 short utterances). However, we noticed that the keyword based method accuracy fell sharply when tested on the larger Twitter dataset we used to train the CNN, yielding only 45% accuracy. We also expect to improve our CNN model in the future training it with more domain specific data, something not possible with a thesaurus based method.

6 Conclusion

In this paper, we have introduced the emotion and sentiment recognition module for an interactive dialog system. We described in detail the two parts involved, namely speech emotion and sentiment recognition, and discussed the results achieved. We have shown how deep learning can be used for such modules in this architecture, ranging from speech recognition, emotion recognition to sentiment recognition from dialogue. More importantly, we have shown that by using a CNN with a single filter, it is possible to obtain real-time performance on speech emotion recognition at 65.7% accuracy, directly from time-domain audio input, bypassing feature engineering. Sentiment analysis with CNN also leads to a 82.5 F-measure when trained from out-of-domain data. This approach of creating emotionally intelligent systems will help future robots to acquire empathy, and therefore rather than committing harm, they can act as friends and caregivers to humans.

Acknowledgments

This work was partially funded by the Hong Kong Phd Fellowship Scheme, and partially by grant #16214415 of the Hong Kong Research Grants Council.

References

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: a cpu and gpu math expression compiler. In *Proceedings of the Python for scientific computing conference (SciPy)*, volume 4, page 3. Austin, TX.

Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Florian Eyben, Martin Wöllmer, and Björn Schuller. 2010. Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1459–1462. ACM.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1:12.

Kaiming He and Jian Sun. 2015. Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5353–5360.

Geoffrey Hinton. 2010. A practical guide to training restricted boltzmann machines. *Momentum*, 9(1):926.

Rie Johnson and Tong Zhang. 2014. Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058*.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Jackson Liscombe, Jennifer Venditti, and Julia Bell Hirschberg. 2003. Classifying subject ratings of emotional speech using acoustic features. In *Proceedings of Eurospeech*, pages 725–728. ISCA.

François Mairesse, Marilyn A Walker, Matthias R Mehl, and Roger K Moore. 2007. Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of artificial intelligence research*, pages 457–500.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

J.W. Pennebaker, R.J. Booth, R.L. Boyd, and M.E. Francis. 2015. Linguistic inquiry and word count: Liwc2015. *Austin, TX: Pennebaker Conglomerates*.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The kaldi speech recognition toolkit. In

IEEE 2011 workshop on automatic speech recognition and understanding, number EPFL-CONF-192584. IEEE Signal Processing Society.

Anthony Rousseau, Paul Deléglise, and Yannick Estève. 2014. Enhancing the ted-lium corpus with selected data for language modeling and more ted talks. In *LREC*, pages 3935–3939.

Paul Schermerhorn and Matthias Scheutz. 2011. Disentangling the effects of robot affect, embodiment, and autonomy on human team members in a mixed-initiative task. In *Proceedings from the International Conference on Advances in Computer-Human Interactions*, pages 236–241.

Björn Schuller, Stefan Steidl, and Anton Batliner. 2009. The interspeech 2009 emotion challenge. In *INTERSPEECH*, volume 2009, pages 312–315. Citeseer.

Björn Schuller, Stefan Steidl, Anton Batliner, Felix Burkhardt, Laurence Devillers, Christian A Müller, and Shrikanth S Narayanan. 2010. The interspeech 2010 paralinguistic challenge. In *INTERSPEECH*, volume 2010, pages 2795–2798.