SPEECH AND TEXT BASED ANALYTICS IN TELUGU LANGUAGE

Rohith Gowtham Kodali, Durga Prasad Manukonda, Rajaraman Sundararajan

rohitkodali@gmail.com, mdp0999@gmail.com, rsundar@auxolabs.com

ASRlytics Inc., Auxo Labs

ABSTRACT

This paper deals with call analytics in Dravidian languages with continuous speech recognition model using Kaldi, Scikit-learn and spaCy. An important part of the speech analytics procedure is to extract the main problem statement along with its answer key and to know the customers opinion about the respective social problems. For speech recognition, we created a standard MFCC for feature extraction for HMM models to align the text and for text analytics. SpaCy NER was used for recognizing the problem and answer keys. For question and answer alignment, Scikit-learn SVM with spaCy model with some user-defined rules were adopted. Sentiment analysis is the second major part of this study and it gives positive-negative calls list by analyzing each sentence of the caller and the responder. Real-time Telugu political calls were collected as data in Andhra region (where Telugu is one of the spoken languages) and was implemented successfully. The goal is to get insights from voice calls and easily convert voice calls into text and analytics from the text for popular languages like Telugu using Multimodal.

Index Terms— Text Analytics, Speech Analytics, Acoustic Modeling, HMM-GMM, LVCSR (Large Vocabulary Continuous Speech Recognition), MFCC, DNN-HMM, Kaldi, IPA, NLP, Multimodal

1. INTRODUCTION

Telugu is spoken by 75 million people and is the third most spoken Indian language by the native speakers. Telugu ranks fifteenth on the list of most-spoken languages worldwide and there are many websites and blogs in Telugu. Currently, government and business organizations are keen in taking customer feedback in this language to improve their services and increase sales. Thus designing NLP tools for Indian languages is crucial for them.

Call analytics is the most efficient analysis technique of call data where, we convert speech to text for identifying insights in a call data. It is used to measure, collect, analyze and generate reports of phone call data. Primarily, business and government organisations use these insights to derive call analyses and optimize their campaigns and call handling process. Organizations use call analytics alongside web analytics [1] to understand which government projects and schemes or advertisements are driving qualified calls, to improve their quality of service and business.

Modern governments interact with people to get feedback about their services. One of the major problems that occur while implementing government schemes and projects. Call analytics is a subset of speech analytics. [2] The most important aspect of call analytics is to know the sectors in which people are suffering with problems like diseases, broken roads, local political leaders, poor services etc. in their regions. The tool in this study is designed to highlight problems and services of government employees and political leaders but these techniques and tools are useful in all business and governance areas. Tele-communication is one of the emerging techniques to interact with people and to get feedback. The goal of implementing call analytics is to measure, manage, and analyze organisational performance efficiently to maximize effectiveness and optimize the return on investment. The key performance indicators in call analytics come from call source (call tracking metrics) and call recording metrics.

Some of the free open-source speech recognition toolkits used for offline research are CMU Sphinx - Speech Recognition Toolkit [3] and Kaldi Toolkit. [4] For this study, Kaldi toolkit is apt as CMU Sphinx gives very low accuracy. Our unique Telugu Speech and Text Analytics [5] Solution is a key component for organization/government and customer-interaction analytics and is the key for unlocking hidden insights, improve customer-satisfaction and faithfulness, high operational efficiency, and high consumer or government performance. The combination of speech-recognition and text analytics is to quickly analyze and categorize 100 percent of voice communications.

The rest of the paper is organized as follows: In section 3, Data from different sources and data-preprocessing. Section 4, the overview of the system architecture and call analysis toolkit. Section 5, system implementation and working process is laid out. Section 6 discusses major points and facts in this toolkit and the future work. Section 7 concludes the study with a brief summary.

2. REVIEW OF LITERATURE

The studies discuss political speech analysis based on existing speech recognition toolkits, with some derived methods. Few of them worked on utterance-based emotion recognition by using efficiency comparison of support vector machines (SVMs) and Binary Support Vector Machines (BSVM) techniques. Frame-based features include acoustic features like energy, mel-frequency cepstral coefficients (MFCC), perceptual linear predictive (PLP), filter bank (FBANK), pitch, their first and second derivatives.[6]

One study focused on Audio and text based multimodal sentiment analysis using features extracted from selective regions and deep neural networks. An improved multimodal approach was proposed, to detect the sentiment of products based on their multimodality natures. Input data was classified as positive or negative or neutral sentiment, along with learning utterance-level representations for speech emotion and age / gender recognition. Accurate recognition of speaker emotion and age / gender from speech can give better user experience for many spoken dialogue systems. [7]

One of the studies, stated that an appropriate mechanism need to be established for conducting sentiment analysis with respect to political debates, the Debate Graph Extraction (DGE) framework was adopted for the study. This framework helps in extracting debate graphs from transcripts of political debates. [8] A real-time SER system based on end-to-end deep learning was proposed by one of the researchers. A Deep Neural Network (DNN) that recognizes emotions from a one second frame of raw speech spectrograms was discussed in detail. It was achieved due to a deep hierarchical architecture, data augmentation, and sensible regularization. For ENTERFACE database and Surrey Audio-Visual Expressed Emotion (SAVEE) database, promising results have been reported. [9]

ACM treats speech recognition as an application of DNNs in its various works. Sahu and Ganesh [10] conducted a survey on HTK, CMU Sphinx and Kaldi toolkits for different languages regarding their performance in terms of WER. They found that Kaldi achieved the best WER value of 2.7 percentage using the Wall Street Journal (WSJ) English corpus. Most of the work done on automatic speech recognition model in the past was based on simple training and decoding of HMM-GMM model. But today, DNNs have proved to be a speedy way for most of the automatic speech recognition systems. DNNs along with HMMs have shown significant improvement over automatic speech recognition tasks. [11] [12] [13][14][15] [16].

One of the studies titled ”Audio and Text based Multimodal Sentiment Analysis using Features Extracted from Selective Regions and Deep Neural Networks”, significantly improved the rate of the sentiment by combining both the modalities such as speech and text[17]. These were the main motives for the research work to develop text-based speech analytics in Telugu. From review of literature, it is evident that this is the seminal study done with Telugu speech analytics engines with the help of spacy and other analytics toolkits. This is the first study to build a domain based Telugu speech analytics engine using the toolkits deep learning approaches.

3. DATA COLLECTION AND DATA PREPROCESSING

3.1. Data Preparation for Kaldi Toolkit

A corpus from Telugu Wikipedia was generated and it was collected using WebBootCaT [18] with seed words of top 2000 high frequency words generated from the Wikipedia corpus. Lexicon was generated for all the words with appropriate phonemes mapping to the Indic script. Then the corpus file was converted into phoneme-level corpus file by changing all the words inside the file with the respective phoneme structures. Sentences with highest triphone sequences were extracted and a training corpus was created. Along with that we added the most common Telugu words like alphabet, number, month names and common nouns. Finally we made a customized version of Woefzela app [19] which can send recorded data to our servers along with speaker-specific information such as age, gender, location. 700 conversational calls related to queries of government projects and schemes were collected and were transcribed manually to be used as trained data for speaker separation task.

3.2. Speaker Separation

At Speech level, 700 calls were collected as data with different accents and at Text level. 3000 caller-spoken sentences and 3000 responder-spoken sentences were taken Additionally, 30 basic questions were framed for identifying caller sentence boundaries.

3.3. Data Preparation for Identification of Key Entities

For training, 6000 sentences were taken from real-time phone calls (Recorded in various parts of Andhra Pradesh, India). A separate app (ASRlytics NER Module) was built to identify and give position numbers to the named entities.

3.4. Data Preparation for SVM

At this stage, 3000 sentences each from caller and responder sentences were taken for identifying the caller and responder. Then sentences taken from recorded phone calls were corrected with the help of a moderator panel in the application to complete the speaker diarization process at text level and speech level.

3.5. Data Preparation for Sentiment Analysis

For collection of data Telugu Wikipedia database dump was used and text was extracted from it using Wikiextractor.py, which can be found at https://github.com/attardi/wikiextractor. Further, the data was preprocessed using our own Python-based text-cleaning scripts. With some amount of manual work top 20000 most frequent words were generated and tagged with their respective Parts-of-Speech tags. Verbs carrying Positive and Negative sentiments, ending up with 1335 negative words and 1051 positive words were identified manually. Some more were added by taking synonyms of these words from grammar books like Shabda Ratnakaram [20]. Text was generated using BootCaT words as seed words. We used the output as our dataset. And from this dataset we further derived these rules which qualify a sentence as positive or negative.

4. SYSTEM ARCHITECTURE

The System Architecture considered for this study is shown below and consists of two modules:

1. Automatic Speech Recognition (ASR)

2. Text Analytics

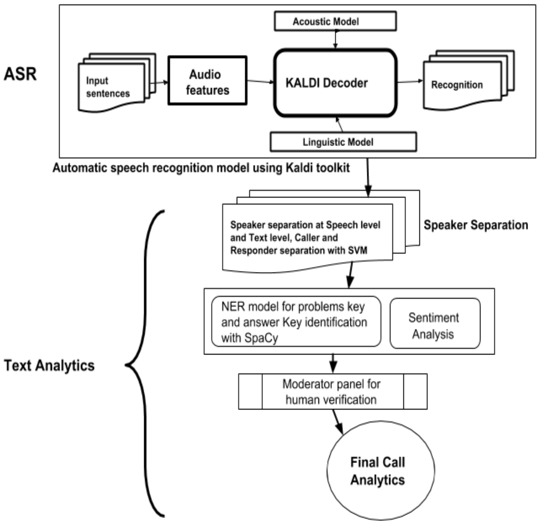

ASR
Acoustic Model
Input sentences
Audio features
KALDI Decoder
Recognition
Linguistic Model
Automatic speech recognition model using Kaldi toolkit
Speaker separation at Speech level and Text level, Caller and Responder separation with SVM
Speaker Separation
Text Analytics
NER model for problems key and answer Key identification with SpaCy
Sentiment Analysis
Moderator panel for human verification
Final Call Analytics

Fig. 1. Call Analysis System Architecture

4.1. Automatic Speech Recognition

Automatic Speech Recognition (ASR) is the process of converting acoustic human speech into text or other symbolic forms of a human language [10]. ASR research focuses on enabling computers to understand human speech and transcribe that speech to text. It is an important field for perfecting intelligent human-machine interaction, machine translation and natural language understanding. Two models are required when training an ASR - a language model (LM) and an acoustic model (AM) [21]. The training process involves:

1. Monophone HMM training with a subset of training data

2. Aligning training dataset using monophone model

3. Triphone HMM training

For offline speech recognition, Kaldi is used to train the speech recognition model. It has several recipes available in it and is an open-source toolkit, written in C++. It has licence under Apache License v2.0. The main goal of Kaldi is to have a modern and flexible code that is easy to understand, modify and extend.

The data preparation process for Kaldi has been explained in section 3.1. After data preparation, MFCC features were extracted from more than 1 lakh phonetically balanced Telugu sentences. Acoustic modeling was performed using GMM and then DNN trained on the labels generated using the GMM models. [22] Decoding was performed using a WFST graph compilation. The performance of both monophone and triphone models using the N-gram language model, computer in terms of word error rate, has been reported. A significant reduction in word error rate was observed when the triphone model was used.

We have used both speech-level and text-level speaker diarization techniques. For speech-level diarization we have used the open source toolkit LIUM SpkDiarization [23] with the existing mode provided. The overall Diarization Error Rate (DER) we have achieved on our conversational calls is 54 percent which is not optimum. So we also used the text based diarization approach.

4.2. Text Analytics

The process of converting unstructured data into meaningful data ready for analysis to measure customer opinions, product reviews, feedback which is commonly referred to as text analytics. Text analytics provides sentiment analysis and entity modeling to enable fact based decision making. Text analysis uses many statistical and machine learning techniques, to determine the keywords, topics, category, semantics and tags from text data.

In our model, we used text analytics to predict key entities of the caller and responder spoken text. The main topics in this area are speaker separation at text level, sentiment analysis, named entities recognition [24] and keywords extraction.

For text-level speaker identification, we used our own code that splits the continuous text to words, grouped together to form sentences based on the caller-and-responder system.

Further this text is corrected by a moderator for ensuring proper sentence formation based on speaker separation. This portion of the code comes under the text analytics part and the techniques used in this area are sentence similarity and Trie-based sentence searching [25].

In speech-level, speaker diarization gave very low accuracy, due to which we introduced this approach to recheck the speaker and caller system to give correct sentences of speakers. The main use of this code is to identify the caller spoken text from training and to leave the responder spoken text. Based on this approach we designed our code to create a caller-and-responder text splitting system. Here is the example operation of caller-spoken text identified from text, from Kaldi (this text has been transliterated using Lekhini, found at http://lekhini.org/nikhile.html, for the reader to understand):

Caller: **halO namastae maDaM!** *(Hello madam!)*
Responder: aa *(Yes?)*
Caller: **maemu pradhaanamaMtri office nuMchi maTlaaDutunnaamu** *(We are speaking from the Prime Minister's office)*
Responder: aa cheppaaMDi maDam. *(Alright, tell me madam.)*
Caller: **prabhutvaM ichchae anni pathakaalu saraina samayaaniki vastunnaayaa saar** *(Are you receiving all the government services on time sir?)*
Responder: aa aa aMdutunnaayi. *(Yes I am receiving all services are good.)*

The bold text is caller-spoken text and is identified by our text separation model and the remaining text is taken as responder-spoken text. This is checked only after text is received from speech-level speaker separation and is used for accuracy of the model at text-level. After sentence splitting, we use Scikit-learn LinearSVC model to classify each sentence into caller or responder [26]. The results and accuracy of this model is discussed in detail in section 6.

Sentiment analysis is a process of classifying sentences as positive, negative or neutral. One major use case of this technology is to find, how people feel about organizations and their products or services via feedback in text format. In call analytics, sentiment analysis is used to know the responder text in positive or negative way. A sentiment analyzer was developed along with defining rules. A rule-based approach is one which uses rules of heuristics to determine sentiments. It uses research in linguistics to analyze sentiments. The main logic of this algorithm has been written based on truth table logic. The results of sentiment analysis in this research is discussed in section 6.

For identification of problem keys, a spaCy NER model [27] was used to identify the main keywords in text and after that sentiment analysis was used to define the number of positive and negative lines expressed in a single call. The detailed analysis shows the number of positive and negative lines spoken by the responder based on the question asked and it extracts the main keywords from the sentiment line, that indicate the problem of the caller and answer key shows why the responder reacted in a positive or negative way. In this tool we have location based call analysis to get a detailed analysis of particular village or mandal. The analysis gives the highest to lowest raised problems by customers or citizens by location.

4.3. Moderator Panel

The moderator panel is used to correct the mistakes made by the machine and it adds corrected text to the training data. The major function of moderator panel is to enable human verification in ASR text and NER keys for higher accuracy and further training the model.

5. SYSTEM IMPLEMENTATION AND WORKING

The aim of this study is to know the perceptions of people about the implementation of government schemes and their services. This feedback helps the government in understanding how their services are utilised and to get suggestions for improvement of services, to bring the problems faced to the notice of government. The significant use of this tool is to get feedback on process of solving peoples problems through phone calls.

The Call Analytics system is a combination of both ASR and Text Analytics systems. The major operations on speech data and data flow at each stage is shown in system architecture (figure 1) and final analytics can be viewed at https://transcript.asrlytics.com:5020/#!/pollytics/location/guntur/nadendla, selecting any of the calls will open the analytics of that call along with the conversation.

246 calls with duration is between 2 to 6 minutes conversation length were taken as test set from Nadendla Mandal (Guntur District, Andhra Pradesh) and remaining calls were used for training purpose. Our model was found to be efficient with higher accuracy. We implemented it for Telugu language but the rules and the code can be implemented in other Dravidian languages like Kannada, Tamil and Malayalam. The test set gives results of call analysis of government in Nadendla Mandal, Guntur as shown in table 1. The calls in the data contain more than one minute of conversation between a caller and responder.

Our system gives the overall sentiment of each call and also the sentiment of each of the conversation lines between the caller and responder. It extracts the key element in question and key answer that decides whether the key is negative or positive. It extracts the keys from positive and negative call conversation lines. One can check complete results of 246 calls of Nadendla mandal in the URI link.

Total benefits are the total number of positive keys (government services and schemes) and unique problems are the negative keys from the answers given by the responders in

ASR	Total Calls	246
Sentiment Analysis	Positive Calls	51
	Negative Calls	98
	Neutral Calls	97
NER and Keys	Total Benefits	227
	Unique Benefits	63
	Total Problems	278
	Unique Problems	74

Table 1. Call Analysis Report of Nadendla, Chilakaluripet and edlapadu Mandals, Andhra Pradesh, India

and around the location of Nadendla, Chilakaluripet and Edlapadu.

The sentiment analysis of a single conversation turn depends on the question asked and the answer given back (Question and Answer system). For example, if the question is negative and the respective answer is also negative, it gives a positive result of the conversation turn. We observed this kind of rule from real time phone calls in Telugu language data. For example:

Conversation Turn 1
Caller: nela nelaa ration correct gaanae istunnaaraa aMDi? (*Are you receiving ration correctly every month?*) **Positive Question**
Responder: istunnaaru aMDi. (*Yes madam, they are giving.*) (Positive Answer) **Positive Question**
Sentiment: **Positive**

Conversation Turn 2
Caller: meeku prabhutvaM nuMDi emainnaa samasyalu unnaayaa aMDi? (*Are you facing any problems from the government?*) **Negative Question**
Responder: laevu aMDi. (*No madam.*) **Negative Answer**
Sentiment: **Positive**

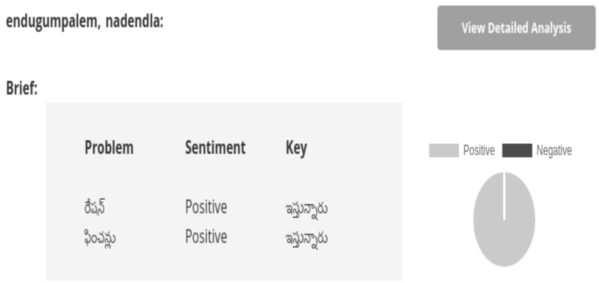
The study considered several rules to determine the sentiment of each conversation turn in the call and later we pass the each negative or positive call to NER model to extract the information for why conversation was negative or positive. For instance above conversation 1 and 2, gives the sentiment positive because the government supplying ration correct time. So, the spaCy NER model results gives below:

Problem Statement: Ration (adopted English word)
Key: istunnaaru *(they are giving)*
(The government is giving the rations properly every month)

Problem Statement: samasyalu (*Problems*)
Key: laevu *(No)*
(It means caller is not having any problems from the government)

Similarly, every call was analysed in each village and reported it at the mandal level. The example of this speech and text based analytics system is shown in Fig.2 and it was single call output. The code logic is working well for Telugu real-time call analytics data with good accuracy. Finally the complete report is shown graphically in Village, Mandal and District level maps in the User Interface.

6. DISCUSSION ON WORK AND RESULTS (DOMAIN SPECIFIC)



endugumpalem, nadendla:

Brief:

Problem	Sentiment	Key
రేషన్	Positive	ఇస్తున్నారు
ఫించన్లు	Positive	ఇస్తున్నారు

Fig. 2. Sample result of single call generated by speech analytics engine

This trial approach has been domain-specific, and focused on calls related to politics and governance. This system was deployed as a beginner working model to take feedback and opinions of citizens regarding government schemes, projects and government services. But the code used is beneficial to be implemented in other domains and is not limited to the domain chosen for this study.

6.1. Kaldi working and accuracy

The data was split into train and test sets, and it was ensured that none of the speakers from the test set are available in the train set. This was done to check how well the model performed for a new speaker. The word coverage of the test set was only 73% of the word coverage in the train set.

The language model and the dictionary that we prepared on the test set were deliberately filled with 5% of Out-Of-Vocabulary (OOV) words. Along with that, 5 different types of noise sounds were inserted, labeled as: cough, laugh, noise, breathe, background noise.

Models were initially trained on MFCC features with first and second derivatives. Then the GMM-HMM system [28] was retrained using LDA+MLLT features. The speaker-adaptive training (SAT) was performed using per-speaker feature-space maximum likelihood linear regression (fMLLR) transforms [29], which is known as LDA+MLLT+SAT. Alignments generated from the SAT model is used to train the BLSTM model [30]. These are the overall models:

Name	Word Error Rate	Misrecognized Words	Insertions	Deletions	Substitutions
Monophone	40.33%	6485 / 16078	230	3130	3125
Triphone	29.65%	4767 / 16078	412	2151	2204
Triphone LDA	29.83%	4796 / 16078	460	2088	2248
Triphone SAT	26.41%	4246 / 16078	578	1638	2030
BLSTM SAT	17.12%	2753 / 16078	362	963	1428

Table 2. Word Error Rate values using various models on the test set. The total number of words are 16078.

1. Monophone
2. Triphone
3. Triphone LDA (Linear Discriminative Analysis)
4. Triphone SAT (Speaker Adaptation Training using FMLLR) on top of LDA alignments
5. BLSTM (Bidirectional Long Short Term Memory)

The corresponding accuracy values on the test set using the models mentioned above can be seen in Table 2. In this table, the total number of insertions, deletions and substitutions have been represented, along with the corresponding accuracies.

6.2. Speaker separation

The combination of speech level and text level speaker diarization gave an accuracy of 79%. This combination was tested using a continuous flow of 800 lines of caller-responder and it predicted 632 as distinct lines. The accuracy is good enough to identify and separate caller and responder from the continuous data.

6.3. Sentiment analysis of caller and responder system

The total number of conversation turns are 2225, taken from 144 distinct calls. Out of these 215 conversation turns were found to be wrongly predicted. Thus the overall accuracy comes out to be 90.33%.

6.4. Main Keyword extraction using NER

The keywords and the answer keys have been extracted only from those calls whose sentiment came out as positive or negative only, and neutral ones were excluded. For 144 phone calls (caller lines = 2341, responder lines = 2331), our spaCy model recognized 313 named entities and 313 key entities in a document during prediction. The actual observations are 390 and the accuracy of the prediction is 80.25%.

7. CONCLUSION

Automatic Speech Recognition was set for the Indian language Telugu, which is a low-level resource language. The data was collected by recording calls on our own for speech and used Wikipedia Telugu dump for text. The linguistic data was collected using WebBootCaT and then processed further. We prepared our phonetics system which mapped letters to sounds. We developed our own POS tagging system with the list of tags taken from C-DAC [31]. After this a corpus was created for Name Entity Recognition (NER), and used spaCy to create NER model for key extraction and answer key identification.

This system is primarily designed as a tool for problem identification in conversational calls. Speaker diarization was tried using LIUM but it did not deliver expected accuracy, so a combination of both LIUM and text-level separation was used, which improved accuracy to a great extent.

This attempt in creating ASR system is one of the first successful trials for a low resource Indian language, with limited datasets available in Telugu language. This successfully implemented a conversational ASR system in the political domain as our first trial, and it can be transferred to any domain with ease.

As discussed in section 6, the combination of speech level and text level speaker diarization gave accuracy of 79%. The prediction of named entities using our NER model brought forth accuracy of 80.25%. The overall accuracy was good (80.06%) but can be improved with training at speech level and rules which apply at text level. To train our models continuously, a web panel was set for linguists to correct the machine-identified NER problem and answer keys, which imparted essential human verification. In addition, the sentences were corrected using a similar panel by various transcriptionists, which was cross-checked by moderators using a moderator panel, allowing the corrected data to be trained for use.

As of now, there are no complete speech analytics multimodels in Telugu language. This is one of the first such successful trials. Further, this system will be implemented in other Dravidian languages.

8. REFERENCES

[1] Avinash Kaushik, *Web Analytics 2.0: The Art of Online Accountability and Science of Customer Centricity*, SYBEX Inc., Alameda, CA, USA, 2009.

[2] Shay Ben-David, A. Roytman, R. Hoory, and Z. Sivan, “Using voice servers for speech analytics,” in *In-*

Module Name	Calls Number	Calls Duration	Accuracy
Kaldi(Speech Recognition)	144	288 minutes	71.00%(Verified manually)
Speaker Separation(Speech+Text)	144	288 minutes	79.00%
Sentiment analysis of caller and responder system	144	288 minutes	90.00%
Information Extraction(Using NER)	144	288 minutes	80.25%
Total Accuracy of Speech Analytics system(speech+Text)	144	288 minutes	80.06%(avg of all modules)

Table 3. Accuracy Table of Multimodel(speech + Text)

ternational Conference on Digital Telecommunications (ICDT'06), Aug 2006, pp. 61–61.

[3] K. F. Lee, H. W. Hon, M. Y. Hwang, S. Mahajan, and R. Reddy, "The sphinx speech recognition system," in *International Conference on Acoustics, Speech, and Signal Processing,*, May 1989, pp. 445–448 vol.1.

[4] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The kaldi speech recognition toolkit," Idiap-RR Idiap-RR-04-2012, Idiap, Rue Marconi 19, Martigny, 1 2012.

[5] Dipanjan Sarkar, *Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from Your Data*, Apress, Berkely, CA, USA, 1st edition, 2016.

[6] N. Kurpukdee, S. Kasuriya, V. Chunwijitra, C. Wutiwiwatchai, and P. Lamsrichan, "A study of support vector machines for emotional speech recognition," in *2017 8th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*, May 2017, pp. 1–6.

[7] Harika Abburi, Suryakanth V. Gangashetty, Manish Shrivastava, and Radhika Mamidi, "Audio and text based multimodal sentiment analysis using features extracted from selective regions and deep neural networks," 2017.

[8] Obinna Onyimadu, Keiichi Nakata, Tony Wilson, David Macken, and Kecheng Liu, "Towards sentiment analysis on parliamentary debates in hansard," in *Revised Selected Papers of the Third Joint International Conference on Semantic Technology - Volume 8388*, Berlin, Heidelberg, 2014, JIST 2013, pp. 48–50, Springer-Verlag.

[9] H. M. Fayek, M. Lech, and L. Cavedon, "Towards real-time speech emotion recognition using deep neural networks," in *2015 9th International Conference on Signal Processing and Communication Systems (ICSPCS)*, Dec 2015, pp. 1–5.

[10] P. K. Sahu and D. S. Ganesh, "A study on automatic speech recognition toolkits," in *2015 International Conference on Microwave, Optical and Communication Engineering (ICMOCE)*, Dec 2015, pp. 365–368.

[11] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov 2012.

[12] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012.

[13] Navdeep Jaitly and Geoffrey Hinton, "Learning a better representation of speech sound waves using restricted boltzmann machines," 06 2011, pp. 5884 – 5887.

[14] Li Deng, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Mike Seltzer, Geoff Zweig, Xiaodong He, Jason Williams, Yifan Gong, and Alex Acero, "Recent advances in deep learning for speech research at microsoft," May 2013, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

[15] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, Jan 2012.

[16] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *2011 IEEE Workshop on Automatic Speech Recognition Understanding*, Dec 2011, pp. 24–29.

[17] Harika Abburi, Suryakanth V. Gangashetty, Manish Shrivastava, and Radhika Mamidi, "Audio and text based multimodal sentiment analysis using features extracted from selective regions and deep neural networks," 2017.

[18] Marco Baroni and Silvia Bernardini, "Bootcat: Bootstrapping corpora and terms from the web," .

[19] Nic J. De Vries, Marelie H. Davel, Jaco Badenhorst, Willem D. Basson, Febe De Wet, Etienne Barnard, and Alta De Waal, "A smartphone-based asr data collection tool for under-resourced languages," *Speech Commun.*, vol. 56, pp. 119–131, Jan. 2014.

[20] B. Sitaramacharyulu, *Sabda ratnakaram: a dictionary of Telugu language*, Asian Educational Services, 1885.

[21] M. Ferretti, G. Maltese, and S. Scarci, "Language model and acoustic model information in probabilistic speech recognition," in *International Conference on Acoustics, Speech, and Signal Processing,*, May 1989, pp. 707–710 vol.2.

[22] Y. G. Thimmaraja and H. S. Jayanna, "Creating language and acoustic models using kaldi to build an automatic speech recognition system for kannada language," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, May 2017, pp. 161–165.

[23] Sylvain Meignier and Teva Merlin, "Lium spkdiarization: an open source toolkit for diarization," in *in CMU SPUD Workshop*, 2010.

[24] Alireza Mansouri, Lilly Suriani Affendey, and Ali Mamat, "Named entity recognition approaches," 2008.

[25] Phil Bagwell, "Fast and space efficient trie searches," 2000.

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

[27] Matthew Honnibal and Ines Montani, "spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing," *To appear*, 2017.

[28] Martin Wöllmer and Björn Schuller, "Enhancing spontaneous speech recognition with blstm features," in *Proceedings of the 5th International Conference on Advances in Nonlinear Speech Processing*, Berlin, Heidelberg, 2011, NOLISP'11, pp. 17–24, Springer-Verlag.

[29] Sree Hari Krishnan Parthasarathi, Björn Hoffmeister, Spyridon Matsoukas, Arindam Mandal, Nikko Strom, and Sri Garimella, "fmllr based feature-space speaker adaptation of dnn acoustic models," in *INTERSPEECH*, 2015.

[30] E. Rodrguez, B. Ruz, . Garca-Crespo, and F. Garca, *Speech/speaker recognition using a HMM/GMM hybrid model*, 1997.

[31] Avinesh PVS and Karthik G, "Part-of-speech tagging and chunking using conditional random fields and transformation based learning," 01 2007.