Practical Applications of Multimedia Retrieval

Language Identification in Audio Files

Tom Herold, Thomas Werkmeister
supervised by Dr. Haojin Yang and Cheng Wang

Hasso Plattner Institute
{tom.herold, thomas.werkmeister}@student.hpi.de

Abstract In this paper we propose a system for language identification in speech audio files using convolutional neural networks. Utilizing Mel-filter spectrogram images obtained from the VoxForge data set and news channels on YouTube we are able to train a robust network architecture for two and four language classifiers. We evaluate our top scoring models on a diverse test set of user-generated speech and music data provided by Dubsmash.

1 Introduction

Recent breakthroughs in audio and image recognition have shown the promise of deep learning models[10]. These models yielded strong improvements in the fields of spoken language identification, machine translation and other language related tasks. The first step in a language processing pipeline is always language identification (LID). Furthermore, many applications arise directly from language identification: 1) customer support centers route people to their native agents, 2) law enforcement officials pinpoint suspects to certain geographic regions and 3) the user-generated content systems rely on accurate feature detection for smart search algorithms.

In this paper we use multi-layer convolutional neural networks (CNN) to approach our research question *Language identification in speech audio files.* Speech audio recognition features well-defined, clear voices and shows very little background noise. This is in contrast to song audio which combines background music, instruments and the singer's voice into a complex mix.

This paper is organized as follows. In chapter 2 we present related work and other approaches to language identification. Chapter 3 gives an overview over the different datasets that were used in our experiments followed by a description of the audio features that serve as input to our system in chapter 4. In chapters 6 and 7 we present our model layer architecture and discuss the performance of different configurations. Chapter 8 features a comparison of Caffe and Tensorflow, the two deep learning frameworks we used. A recommendation for future work is presented in chapter 9 and a summary of our key findings can be found in chapter 10

2 Related Work

Several approaches and data sets exist for language identification. In this section our focus are recent works using deep learning techniques.

Garofolo et al. published the TIMIT [3], a corpus of phonemically and lexically transcribed speech of American English speakers of different sexes and dialects.

The TopCoder Spoken Languages Challenge[15] provides speech audio files for 176 languages. The dataset comprises 66,176 mp3 files, each of which contains approximately 10 second of speech recorded. On average this yields 376 samples per language.

Srivastava et al. [14] use a language independent phoneme classifier to extract the sequence of phonemes from the audio. In the next step this phoneme sequence is classified using statistical and recurrent neural network models.

Montavon[12] proposes a deep (3 convolutional layer) and shallow (1 convolutional layer) time-delay neural network (TDNN) architecture for language identification. Using English, German and French he achieves 91.3% accuracy for known-speakers and just 801% for new speakers during evaluation. He relies on the open source Voxforge speech sample dataset and a samples from various radio stations converted to 5 second long spectrograms featuring 39 mel buckets.

Lee et al. [11] use unsupervised convolutional deep belief networks (CDBN) to learn phonemes from speech data. They create 20ms spectrograms with 10ms overlaps from the English-only TIMIT corpus. The spectrograms were further processed using PCA whitening to reduce the dimensionality. As a second research task they use unsupervised CDBNs for gender identification in audio files.

Graves et al. [4] use recurrent neural network (RNN) for language identification. They found that deep long short-term memory (LSTM) RNNs [6] achieve a test set error of 17.7% on the TIMIT phoneme recognition benchmark.

3 Datasets

We focus on four languages for our research: English, German, French and Spanish. We train our models on data from two data sources: speech samples from VoxForge and news reels from YouTube. We evaluate the performance of our models on a small real-world test set consisting of user-uploaded content from the Dubsmash platform.

3.1 VoxForge

VoxForge[1] is an open-source speech audio corpus containing samples from 18 different languages. With 6000 samples the English language set accounts for the largest share. Table 1 shows an overview of sample sizes per language. The data set consists of short user uploaded audio files together with meta data and a machine transcription of the spoken text. All samples are about 5-10

[1] http://www.voxforge.org/

seconds long and have varying speakers, resulting in a total duration of 1.5h - 6.5h per language. Audio quality between different samples varies based on the recording equipment used by the speaker. In general, the audio quality is high.

The speech is very characteristic across all files. Each file contains only one speaker reading text. The resulting speech is slowly paced, well pronounced and very audible. Therefore, the data set sounds rather artificial and is very different from regular speech between two persons.

3.2 YouTube

In order to increase our data set size, we downloaded large amounts of news reel videos from YouTube[2]. Popular channels like CNN contain several hundred videos, most of which are several minutes up to an hour long. We obtained about 40 hours of audio data per language. Table 2 shows a complete overview of involved news channels. In the future, our data set can be extended by sourcing more speech data from YouTube. Also, other languages or songs can be added to the mix.

In contrast to the artificial nature of VoxForge the YouTube data set usually has several persons speaking to one another. The resulting speech sounds more natural and is paced at a medium-speed to fast level. The recording quality is very high and free of noise. Although most of the content has no background noise, occasionally news stories feature clips about events or products that do not contain speech data.

3.3 Dubsmash

We validate our models on a user-generated data set with 3000 samples for our four languages from Dubsmash[3]. Using the Dubsmash app, users can choose an audio recording or soundbite from movies, shows, music, and internet trends and record a video of themselves dubbing over that piece of audio. This data is very different from the well-defined audio which were recorded in a mostly professional environment on which we train our models.

The Dubsmash soundbites are both curated and user-uploaded. Hence the quality and content varies greatly. The data set contains both very poor recordings of radio songs created using smart phones as well as professionally created speech files. Most files come with background noise, either from poor recording gear or setting, or from intended background music. About 25% of all files are songs.

[2] http://www.youtube.com

[3] http://www.dubsmash.com

3.4 Data Set Split

For both VoxForge and YouTube, our training data sets, we use an 80% / 20% test split and an even sample distribution among languages. Through preprocessing we generate 30.000 images of padded, five second long, non-overlapping sound snippets for the YouTube data set.

Table 1. VoxForge language distribution

Language	Number of Samples
English	6000
German	1300
French	1800
Spanish	2200

Table 2. YouTube Data Sources

Language	YouTube Channel Name
English	CNN, bbcnews
German	deutschewelle, euronewsde, N24de
French	france24, myFrancetv
Spanish	antena3noticias, rtve

4 Features

In the following, we present possible features that can be extracted from audio files for the purpose of machine learning. Due to the temporal variance in audio signals, the raw amplitude measurements over time, also called waveform, are often transformed into different representations. Only lately researchers started working with the raw waveform [13]. Figure 1 depicts the raw waveform of an audio signal.

In many of these representation the Discrete Fourier Transformation (DFT) plays an important role. The DFT analyzes a window of amplitude measurements from the waveform and calculates the strength of frequencies inside that window.

4.1 Spectrograms

Spectrograms are a visual representation of the audio signal after the application of the DFT. Figure 2 shows a spectrogram. Spectrograms show patterns that are visible to the human eye. These patterns are lines that are flat, rise, or drop over time and are usually present in multiple frequencies. In related work these patterns were used to recognize phonemes [11]. The human voice mostly operates in frequencies below 10kHz [9]. Thus, higher frequencies can be truncated.

4.2 Mel Filtering

Spectrograms contain a lot of redundancy and are large in size. Lee et al. [11] use PCA whitening to reduce the size of this representation. Another method is Mel-filtering. Mel-filtering collects frequencies into a specified amount of frequency buckets. Below 1kHz the buckets are spaced linearly, while above 1kHz they are

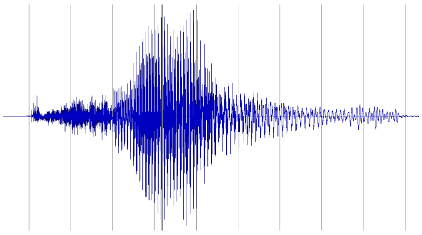

Figure 1. Raw waveform of an audio signal. The horizontal axis is the time and the vertical axis is the measured amplitude at this point in time

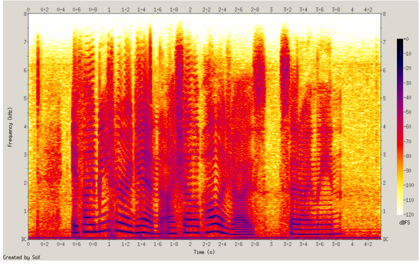

Figure 2. Spectrogram of an audio signal. The horizontal axis is the time and the vertical axis is the frequency. The darker the color the stronger the frequency at the given time

spaced logarithmically. This modeling corresponds to human hearing capabilities that are more fine grained below 1kHz and are less able to distinguish differences in higher frequencies [9]. In related work the number of buckets chosen was 39 [12].

4.3 MFCC Vectors

In contrast to the prior features, the Mel frequency cep- stral coefficients (MFCC) are not a visual representation of the audio signal. Instead, they capture information about the audio signal in a 39 dimensional feature vector. By applying the DFT, mel filtering, and the inverse DFT the 39 coefficients can be extracted. Jurafsky and Martin describe the process in detail [9].

5 Preprocessing

We process the audio files in multiple steps to generate a visual representation. First, we split audio files into 5 second pieces without overlap. We then create spectrograms and apply Mel-filtering. To reduce noise in the resulting images, we apply a levels filter that maps darker gray values to black. Finally we pad every image to 600 pixel width appending zeros to the right end. Figure 3 shows the intermediate results of the preprocessing steps.

6 Models

In this paper we propose two different convolutional neural network models: a shallow and a deep architecture. The shallow network consists of three convolutional layers (kernel 6x6, stride 1), each followed by a max pooling layer (kernel 2x2, stride 2) and with either local response normalization (LRN) or batch normalization (BN). (More on the performance impact of the two in chapter

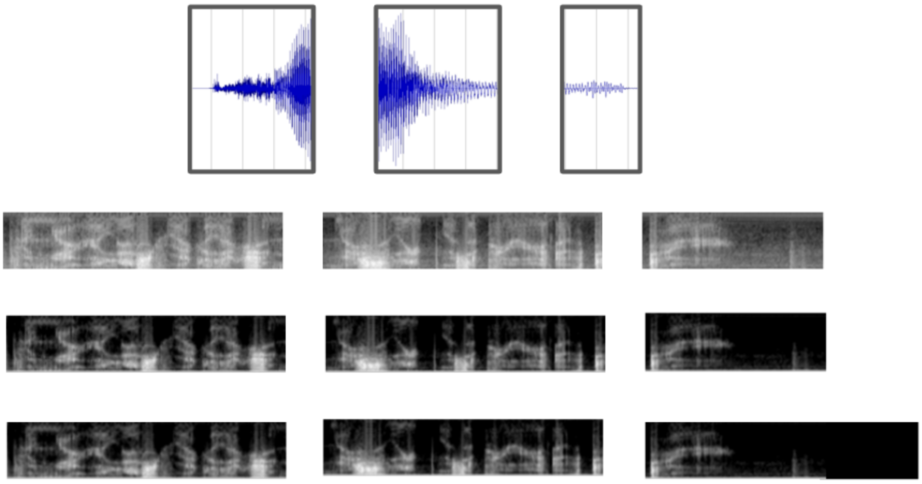

Figure 3. Feature Transformation: Top to bottom. 1) Wave Form 2) 5 second splits 3) Mel filter image 4) Noise-reduction with levels filter 5) Padded images.

7.) The network features two fully connected layers after the convolutions and results in two or four labels, depending on the number of languages used for training. Layers are activated using the ReLU function. Figure 4 visualizes the architecture of the shallow net. This shallow architecture yields a top accuracy of 92.9% for two languages using Mel-filter images. (See chapter 7 for details.)

We also evaluated a deep model architecture with six convolutional, three fully-connected and one dropout layer. figure 5 shows the complete architecture. Harutyunyan proposed a similar architecture for use with raw spectrogram images [5] . It turns out that this deep model does not work for the relatively narrow Mel-filter images (39px in height) since the pooling layers shrinks the images too much.

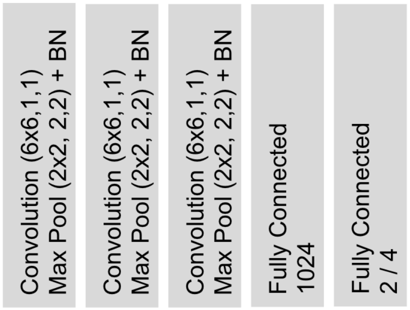

Figure 4. Shallow Network Architecture

Comparing the two approaches, we achieve 83.06% accuracy using raw spectrogram approach and the deep model, and 84.53% accuracy for Mel-filtered images using the shallow model for four languages.

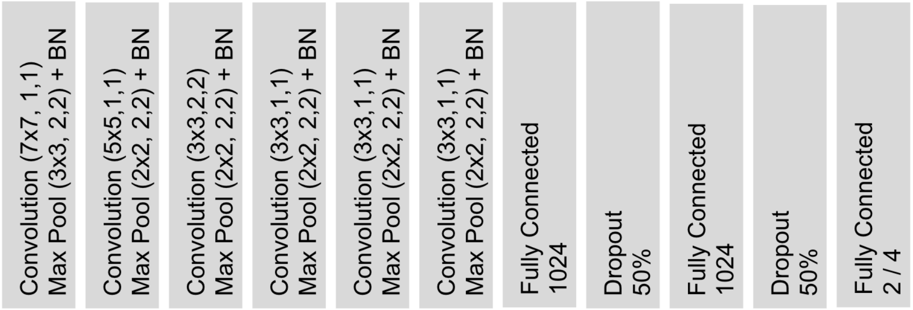

Figure 5. Deep Network Architecture

7 Results

In this chapter we list the accuracy of our top scoring networks on two different data set. All models were trained on the YouTube data set (see 3.2), so we optimized the network performance against it. To evaluate the robustness and generalization of the models we feed them with the Dubsmash data set.

7.1 Evaluation on YouTube Data Set

We utilize the Caffe framework[8] to train both the shallow and deep network architectures. Using stochastic gradient descent we achieve a top accuracy of 92.9% and 84.5% for two and four languages respectively on the YouTube data set. We achieve significant improvements using two methods. Firstly, applying a levels-filter for noise reduction in the Mel-filtered spectrogram images boosts training accuracy. Secondly, Ioffe et al. proposed adding a Batch Normalization (BN) layer [7] after each convolution layer. Swapping local response normalization (LRN) layers with BN layers yields up to 5% improvement in accuracy for the four language setup. Table 3 shows a complete overview of our results. The top scoring networks were trained with 15.000 images per languages, a batch size of 64, and a learning rate of 0.001 that was decayed to 0.0001 after 7.000 iterations.

Previous work[12] raised concerns about similarity of the feature languages and the networks inability to discriminate between them properly. After all English and German both belong to the West Germanic language family, while French and Spanish are part of the Romance languages. We are positive, however, that our deep learning approach to language identification was able to differentiate

Table 3. Top accuracy for two / four languages using modification of the shallow architecture.

Accuracy	Dataset	Net
0.929	Youtube DE / EN	3 Conv + BN + 2 FC
0.908	Youtube DE / EN	3 Conv + LRN + 2 FC
0.845	Youtube DE / EN / ES / FR	3 Conv + BN + 2 FC
0.789	Youtube DE / EN / FR / ES	3 Conv + LRN + 2 FC
0.800	Voxforge DE / EN / FR	3 Conv (Montavon Paper[12])

between them reliably. Table 4 shows the error rate when evaluating language family pairs on the best performing four language model. Spanish and French audio files separate very well with hardly any wrong classifications. German exhibits a similar performance level. Only English language samples fare slightly worse and are classified as German or surprisingly as French.

Table 4. Performance of the language family discrimination on known and unknown speakers. Rows of the confusion matrices represent the true label and columns represent the prediction of the classifier. Total Accuracy 91.3%.

	EN	DE	ES	FR
EN	0.173	0.03	0.011	0.007
DE	0.003	0.32	0.004	0.002
FR	0.0006	0.003	0.21	0.004
ES	0.0004	0.002	0.005	0.21

7.2 Evaluation on Dubsmash Data Set

To evaluate the robustness and real life performance of the two top networks we tested them on the Dubsmash data set. The data was processed in the same manner as the YouTube training data set. We experienced a significant drop in accuracy compared to our evaluation of the YouTube data set. This can be explained by the different nature of the Dubsmash audio files. (see table 5) Many of the Dubsmash sounds contain background noise ranging from static noise of a poor recording all the way to full background music. Some files are complete songs, a situation not covered in our the training material. Furthermore, some samples feature the use of slang words and sound distortions. E.g. a loud cries of "Yooo daawwg" is just as complicate to recognize as child and baby voices. 5

Table 5. Evaluation on the Dubsmash data set.

Accuracy	Languages	Net
0.626	DE / EN	3 Conv + BN + 2 FC
0.439	DE / EN / FR / ES	3 Conv + BN + 2 FC

8 Frameworks

We run our experiments both using the popular Caffe[8] and the recently released Tensorflow[1] deep learning frameworks. While both systems feature convolutional neural network training, their approach to it is different. In this chapter we describe our observations working with the two different frameworks.

Caffe can be configured through of declarative *.prototxt model files and can be run from the command line. Tensorflow, however, can only be accessed through C++ and Python. To use Tensorflow a lot of boilerplate code is needed to define network architectures and the learning environment. While this enables fine-grained configurations, it requires much more programming and deep learning experience. We found that Caffe outperforms Tensorflow significantly on our development server with a Nvidia TitanX with 12GB of VRAM. With Tensorflow we achieve a throughput of 50 images/second and 280 images/second with Caffe. Similar observations hold for memory consumption. Tensorflow was unable to load a nine layer deep network on our machine due to insufficient GPU memory. In contrast, Caffe was able to load and work with the deep model.

Chintala et al. [2] created a benchmark of various popular deep learning frameworks. Table 6 shows a speed comparison of these running a single forward and backward pass using different published networks. Caffe is faster in two out of three model trainings, which conforms with our findings.

8.1 Web Demo

We implemented a web demo showcasing our system. The demo allows users to upload sound files and returns the individual probabilities for each language.

The demo web server is using the Flask[4] framework for Python and loads the model and makes a prediction upon user upload.

9 Future Work

While there are many possible improvements to our system, two branches of future work seem most promising: the augmentation of our audio data and the use of recurrent neural networks.

[4] http://flask.pocoo.org/

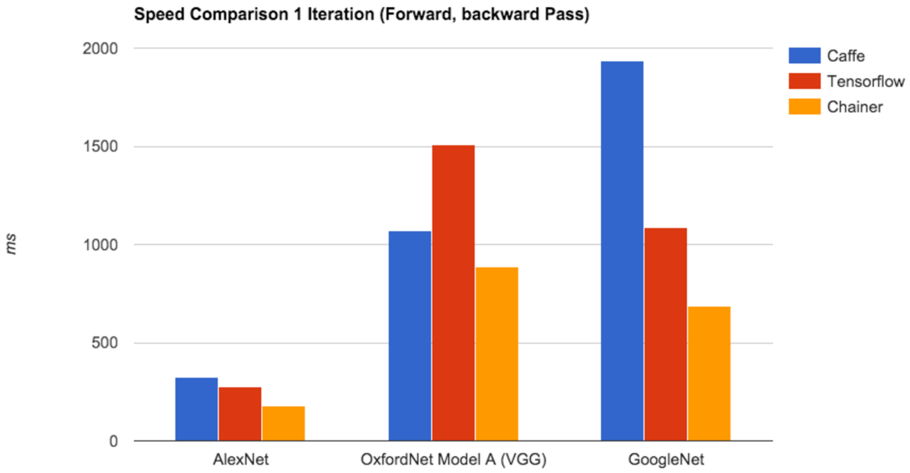

Figure 6. Speed Comparison of popular Deep Learning Frameworks.

9.1 Data Augmentation

Data Augmentation refers to additional steps in the preprocessing pipeline that increase the number of training samples and thus the robustness of the learned model.

As a first step, an overlap should be introduced to the five second splits of the audio files. Using a step size of one second to create the snippets would result in a 5-fold increase of training samples.

As a next step, background noise can be introduced to help the model better generalize over different audio qualities or environments the speaker is in.

Finally, the voice of the speaker can be altered by lowering or heightening the pitch creating robustness for a wide range of speakers interacting with the system.

9.2 Recurrent Neural Networks

This branch of future work deals with a different architecture for processing audio data. So far, we used convolutional neural networks to process visual representations of audio data. In these networks the temporal aspect of the data is captured using a long extract from the audio file. Recurrent neural networks on the other hand, naturally work with sequential data. They accept arbitrarily long input sequence and return a single classification result.

To stay close to our current architecture, a recurrent layer could be introduced behind the convolutions and before the fully connected layers. Audio files could be split up into smaller chunks of one second or less and bed fed into the network sequentially. This would allow us to process different audio lengths without the need for extensive padding at the end of the 5 second snippets. Secondly, it directly creates a classification result for longer sequences, eliminating the need for combining classifications of multiple snippets. Additionally, recurrent neural networks, specifically Long Short Term Memory (LSTM) networks, deliver state of the art performance in speech recognition [4].

10 Summary

We replicated prior results in language identification [12] on the voxforge and news channel audio. We applied our trained models to novel data provided by Dubsmash and found that accuracy drops by 30% in the case of two languages and by 40% for four languages. Given the broad domain and diversity of the Dubsmash audio files, these results are not surprising. Reliable training data is difficult to obtain for such a classification of these diverse audio files. Hence, we assume that the best performance can be obtained by combining our language identification system with other classifiers using ensemble methods, such as boosting. Dubsmash could profit from this approach, because they also have information about the location or interface language of a user.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), http://tensorflow.org/, software available from tensorflow.org
2. Chintala, S.: Convnet benchmarks (2015)
3. Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S.: Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. NASA STI/Recon Technical Report N 93 (1993)
4. Graves, A., Mohamed, A.r., Hinton, G.: Speech recognition with deep recurrent neural networks. In: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. pp. 6645–6649. IEEE (2013)
5. Harutyunyan, H.: Spoken language identification with deep convolutional networks (2015), https://yerevann.github.io/2015/10/11/spoken-language-identification-with-deep-convolutional-networks/, [Online; accessed 29-March-2016]
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
7. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
8. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia. pp. 675–678. ACM (2014)

9. Jurafsky, D., Martin, J.H.: Speech and Language Processing (2Nd Edition). Prentice-Hall, Inc., Upper Saddle River, NJ, USA (2009)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
11. Lee, H., Pham, P., Largman, Y., Ng, A.Y.: Unsupervised feature learning for audio classification using convolutional deep belief networks. In: Advances in neural information processing systems. pp. 1096–1104 (2009)
12. Montavon, G.: Deep learning for spoken language identification. In: NIPS Workshop on deep learning for speech recognition and related applications. pp. 1–4 (2009)
13. Sainath, T.N., Weiss, R.J., Senior, A.W., Wilson, K.W., Vinyals, O.: Learning the speech front-end with raw waveform cldnns. In: INTERSPEECH. pp. 1–5. ISCA (2015), http://dblp.uni-trier.de/db/conf/interspeech/interspeech2015.html#SainathWSWV15
14. Srivastava, B.M.L., Vydana, H.K., Vuppala, A.K., Shrivastava, M.: A language model based approach towards large scale and lightweight language identification systems. CoRR abs/1510.03602 (2015), http://arxiv.org/abs/1510.03602
15. TopCoder Inc.: Topcoder spoken languages challenge 2 (2010), https://community.topcoder.com/longcontest/?module=ViewProblemStatement&rd=16555&pm=13978, [Online; accessed 10-March-2016]