SVitchboard II and FiSVer I: High-Quality Limited-Complexity Corpora of Conversational English Speech

Yuzong Liu, Rishabh Iyer, Katrin Kirchhoff, Jeff Bilmes

University of Washington, Seattle

{yzliu,rkiyer,kk2,bilmes}@uw.edu

10.21437/Interspeech.2015-238

Abstract

In this paper, we introduce a set of benchmark corpora of conversational English speech derived from the Switchboard-I and Fisher datasets. Traditional ASR research requires considerable computational resources and has slow experimental turnaround times. Our goal is to introduce these new datasets to researchers in the ASR and machine learning communities (especially in academia), in order to facilitate the development of novel acoustic modeling techniques on smaller but acoustically rich corpora. We select these corpora to maximize an acoustic quality criterion while limiting the vocabulary size (from 10 words up to 10,000 words) with different state-of-the-art submodular function optimization algorithms. We provide baseline word recognition results for both GMM and DNN-based systems and release the corpora definitions and Kaldi training recipes to the public.

Index Terms: speech recognition, acoustic modeling, submodular optimization

1. Introduction

Speech recognition is one of the most challenging tasks in applied machine learning, and one that requires enormous amounts of rich training data. Among different aspects of a speech recognition system, training acoustic models for conversational speech recognition is one of the most challenging tasks. First, the acoustic characteristics of conversational speech are more diverse than those of carefully read speech due to increased variability in pronunciation, speaker, and environment. Second, conversational speech recognition involves large vocabularies. Thus, a very large amount of training data is required to train a conversational speech recognition system. Finally, recently developed acoustic modeling techniques using deep architectures [1, 2, 3, 4, 5, 6, 7], require long training and, thus, system development times.

While conversational speech recognition is challenging, it is even more difficult for researchers with limited computational resources. The complexity of acoustic model training is usually linear in n (i.e., $O(n)$), where n is the number of tokens in the training data. For very large n and computationally demanding models like DNNs, it can take weeks to train just one system even on GPUs. Such long experimental turnaround times makes large-scale speech recognition impractical, particularly in academia where most researchers and students have limited computational resources. Even outside of academia, this problem limits the evaluation of many diverse models since fewer models can be evaluated given a fixed time and compute budget.

Two of the most commonly used conversational speech corpora are the Switchboard [8] and Fisher [9] datasets, both of which are large in terms of vocabulary size and number of training samples. Our goal is (a) to produce useful but acoustically rich subsets of Switchboard and Fisher, (b) to establish baselines performance numbers, and (c) to release the corpora definitions for free to the community. We refer to the resulting corpora as SVitchboard-II (SVB-II), and FiSVer-I, where in each case "SV" stands for "small vocabulary." By doing so, we hope to provide researchers with smaller but still challenging speech corpora, thus facilitating faster experimental throughput for testing novel acoustic modeling and machine learning methods.

2. Goals

The basic goal of **high-quality limited-complexity corpus selection** is to choose a large subset X of a *ground set* V of speech utterances (e.g., the entire 309-hour Switchboard-I dataset) that has limited complexity but is similar to the original dataset in some way. That is, we wish to choose a subset $X \subseteq V$ that have the following two properties:

1. **high quality:** That is, X being high quality might mean the utterances X constitute a large amount of speech, a large number of tokens, or be acoustically diverse and/or confusable in some way. We construct a function $g(X)$ that measures the quality of X, and we choose X such that $g(X)$ is maximized.
2. **low complexity:** Complexity may correspond to computational cost, so an obvious complexity measure might be the vocabulary size in X (i.e., the number of distinct types in X). We define a function $f(X)$ that measures the complexity of X, and choose X such that $f(X)$ is minimized.

In a previous study [10], a heuristic was proposed to select different subsets of Switchboard (with vocabulary size of 10, 25, 50, 100, 250, and 500 words). The resulting corpora, named "SVitchboard I", are available online at `http://tinyurl.com/svitchboardI`. The heuristic greedily selected the most frequent words in the transcripts until the vocabulary size constraint were met, a procedure that can have unboundedly poor performance [11] since it implicitly defines a supermodular function that is maximized via the greedy method. In our work, we investigate a principled approach to data selection using submodular function [12] optimization. Our approach builds upon [11] where a subclass of the algorithms we present here was considered in [11] for subselecting data. However, [11] only proposed and tested subselection algorithms for this problem, but it did not produce experimental speech recognition results or provide resulting corpora definitions. Here, we consider a more general class of algorithms (that includes those proposed in [11]) and use them to find the best *corpus* in terms of various statistics. Furthermore, we run baseline GMM-HMM and DNN based ASR systems on these corpora.

3. Submodular Optimization

We formulate the problem of selecting a high-quality, limited-complexity corpus as a submodular function optimization

problem. Submodular functions are set functions that have the 'diminishing returns' property. Given a finite set V, a set function $f : 2^V \rightarrow \mathbb{R}$ is said to be submodular if $f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B)$ holds $\forall A \subseteq B \subseteq V$ and $v \notin B$. I.e., the incremental value (or 'gain') of element v decreases as the context in which v is considered grows from A to B. We also define *modular* functions as those that satisfy the above inequality everywhere with equality. Submodular functions have shown strong performance in several real world applications such as feature selection [13, 14, 15, 16], clustering [17], structure learning [18], document summarization [19, 20], image collection summarization [21], speech training data selection [22, 23], sensor placement [24], and many others. In this paper we show that several natural instantiations of the quality function g and the complexity function f are submodular, thereby providing a principled approach to data subselection by simultaneously minimizing a submodular function (complexity) while maximizing another (quality).

3.1. Constrained Submodular Optimization

[25] defined a number of algorithms to solve the following two constrained submodular optimization problems, referred to as "Submodular Cost Submodular Cover (SCSC)", and "Submodular Cost Submodular Knapsack" (SCSK), respectively:

$$\text{Problem 1 (SCSC): } \min\{f(X) \,|\, g(X) \geq c\}, \text{and}$$

$$\text{Problem 2 (SCSK): } \max\{g(X) \,|\, f(X) \leq b\},$$

where both $g : 2^V \rightarrow \mathbb{R}_+$ and $f : 2^V \rightarrow \mathbb{R}_+$ are polymatroid (non-negative monotone-nondecreasing submodular) functions.

This addresses exactly the problem we wish to solve. In particular, we can use the formulation of Problem 2 and directly enforce constraints on the vocabulary size while maximizing the quality. Unlike submodular function minimization, however, this problem is NP-hard [25]. Several of the algorithms proposed in [25], however, are scalable and admit bounded approximation guarantees. In this paper, we use the iterative submodular knapsack algorithm, outlined in Section 4.2 in [25].

3.2. Difference of Submodular Functions Optimization

The second approach we consider minimizes the difference between submodular functions [26, 27]:

$$\text{Problem 3 (DS): } \min_{X \subseteq V} v(X) \tag{1}$$

where $v(X) = \lambda f(X) - g(X)$ is a difference of two submodular functions. Similar to SCSC/SCSK, this method addresses the underlying problem; different values of λ will amount to different vocabulary sizes. Unfortunately, unlike SCSC and SCSK, we do not have explicit control over the vocabulary size and we instead need to tune λ to obtain the right solution. Like SCSC/SCSK this problem is NP-hard, but the algorithms proposed in [26, 27] are scalable and work well in practice.

3.3. Unconstrained submodular function minimization

The third (and final) approach we consider is

$$\text{Problem 4 (SFM): } \min_{X \subseteq V} h(X)$$

where $h(X) = g(V \backslash X) + \lambda f(X)$ is a submodular function. We minimize $g(V \backslash X)$, the quality of X's complement $V \backslash X$, rather than maximizing the quality of X. Since $g(\cdot)$ is polymatroidal and normalized, we have $g(V \setminus X) \geq g(V) - g(X)$. Thus, we minimize an upper bound $(g(V \setminus X) + \lambda f(X))$ of the objective rather than the actual objective $(g(V) - g(X) + \lambda f(X))$. However, if $g(\cdot)$ is modular, then we exactly maximize the objective [11]. Also, $h(X)$ with $\lambda \geq 0$ is a mixture of two submodular functions; hence, $h(X)$ is also submodular and we can minimize h exactly using unconstrained submodular function minimization in polynomial time. Third, finding solutions for **all** possible values of λ and finding solutions for a **single** λ have the same complexity, thanks to the **principle partition** of submodular systems (see [12, 28]). On the other hand, this approach has the disadvantage that for strictly submodular $g(\cdot)$ we minimize only an upper bound of our goal. Another disadvantage is that we have to accept the solutions that we get for different values of λ, and in general there is only a small set of critical values of λ that matter — any other value of lambda will produce a solution that is identical to one of the critical values of λ. This is a disadvantage if the resulting solutions do not fit our goals, needs, and budgets. Finally, for submodular g and f, general purpose submodular function minimization, while theoretically requiring polynomial time, can be slow in practice.

3.4. Corpus creation via various $g(\cdot)$ and $f(\cdot)$

We next describe different function instantiations for $g(\cdot)$ and $f(\cdot)$. We start with four different modular functions as quality functions g. All are normalized so that $g(\emptyset) = 0$ and $g(V) = 1$.

Utterance count: $g_1(X) = |X|/|V|$. This defines high quality as containing a large percentage of utterances in V. Each utterance (short or long) is given equal weight.

Amount of speech: $g_2(X) = w_V(X)/w_V(V)$ where $w_V(X) = \sum_{v \in X} w_V(v)$ and $w_V(v)$ measures how much speech (excluding silence) is in the acoustic signal v.

Number of tokens: $g_3(X) = w_V(X)/w_V(V)$ where $w_V(X) = \sum_{v \in X} w_V(v)$ and $w_V(v)$ measures how many tokens are contained in the transcription of utterance v.

Intra-utterance acoustic dispersion/diversity. $g_4(X) = w_V(X)/w_V(V)$ where $w_V(X) = \sum_{v \in X} w_V(v)$ and $w_V(v)$ measures the "acoustic dispersion" of utterance v. If $x^v = (x_1^v, x_2^v, \ldots, x_T^v)$ is a sequence of MFCC vectors for utterance v, then we can measure acoustic dispersion via:

$$w_V(v) = \frac{1}{T^2} \left| \sum_{i=1}^{T} \sum_{j=1}^{T} (x_i^v - x_j^v)(x_i^v - x_j^v)^\intercal \right| \tag{2}$$

Hence, we prefer an utterance if it is acoustically diverse.

All the above functions are modular, i.e. the score of an utterance does not interact with the score of another. Thus, there is a high chance of choosing a set X that has high quality but that is also redundant. As an extreme example, if a corpus had duplicate entries each of which is very high quality, both would be chosen even though the corpus diversity would not improve.

To address this problem we utilize a strictly submodular function for $g(\cdot)$ in order to choose not only high-quality but also a diverse set of utterances. A natural choice for g is the class of feature based function [23, 29, 30], defined as,

$$g_5(X) = \sum_{u \in \mathcal{U}} w_u \phi(m_u(X)) \tag{3}$$

where $\phi(\cdot)$ is a non-negative monotone non-decreasing concave function, $\mathcal{U}$ is a set of features, w_u is a non-negative weight of feature u, and $m_u(S) = \sum_{j \in S} m_u(j)$ is a non-negative score for feature u in set S, with $m_u(j)$ measuring the degree to which utterance $j \in S$ possesses feature u. Each term in Eq. (3) is based on a "feature" or "concept" of the objects X being scored. Features can consist of phonetic or prosodic feature labels (e.g. phonemes, triphones, words, syllables, tones, paralinguistic attributes, etc.) and hence, such functions are useful

SVitchboard-II Dataset									
Task	Vocab Size	Avg. Phone	# Utts	# Tokens	Speech (hrs)	g-value	# conv.	norm. ent 1	norm. ent 2
50	50	3.32	24033	38154	4.01	4.13054e9	4491	0.4688	0.3916
100	100	3.28	27228	51254	4.93	4.8425e9	4571	0.4998	0.4203
500	500	3.95	39694	131815	10.30	7.70767e9	4749	0.6122	0.5243
1000	1001	4.50	48445	230876	16.81	9.70981e9	4801	0.6831	0.5911
5000	5003	5.55	74162	668261	46.28	1.49496e10	4867	0.78340	0.6911
10000	9983	5.97	84636	883710	61.19	1.68402e10	4871	0.8059	0.7152
All	30021	6.22	262473	3109768	224.11 (310 total)	1.0244404e11	4876	0.8016	0.7108
FiSVer-I Dataset									
Task	Vocab Size	Avg. Phone	# Utts	# Tokens	Speech (hrs)	g-value	# conv.	norm. ent 1	norm. ent 2
10	10	4.6	64998	73650	9.96	3.21993e10	15561	0.4740	0.3815
50	50	5.92	115512	144906	17.95	6.91023e10	21052	0.5609	0.4678
100	100	5.6	138722	191156	22.01	9.56028e10	22062	0.5891	0.4958
500	500	5.34	258307	653847	55.32	2.25651e11	23111	0.7129	0.6175
1000	1000	5.43	352261	1299566	99.06	3.23534e11	23214	0.7744	0.5911
All	42154	6.36	1.7M	17M	1242.5 (1593 total)	1.30739e12	23300	0.8596	0.7737

Table 1: Statistics of SVitchboard-II (top table) and FiSVer-I (bottom table) datasets. Vocab size: actual vocabulary size; Avg. Phone: average number of phonemes per word; #Utts: number of utterances; #Tokens: number of tokens; Speech: hours of speech (excluding the silence parts); g-value: the function value of $g_5(X)$; # conv.: number of conversation sides; norm. ent 1: normalized entropy of phoneme distribution; norm. ent 2: normalized entropy of non-silence phoneme distribution

for applications in speech processing. Maximizing this objective asks for sets X that possess diversity over and coverage of the features. Such functions, moreover, easily scale to large-scale data selection problems.

In this work, $\mathcal{U}$ is the set of clustered triphone HMM state labels produced by a forced Viterbi alignment of the word transcriptions (using a trained system), and $\phi()$ is the square root function. The score $m_u(s)$ is the count of feature u in element s, normalized by term frequency-inverse document frequency (TF-IDF), i.e., $m_u(s) = \text{TF}_u(s) \times \text{IDF}_u(s)$, where $\text{TF}_u(s)$ is the count of feature u in s, and $\text{IDF}_u = \log(\frac{|V|}{d(u)})$ is the inverse document count of the feature u with $d(u)$ being the number of utterances that contain the feature u (each utterance is considered a "document"). The weight for u is given as $w_u = m_u(V)$.

One obvious candidate for a submodular complexity function $f(\cdot)$ is the vocabulary corresponding to X. We define $f(X) = w_U(\gamma(X)) = \sum_{u \in \gamma(X)} w_U(u)$ where $\gamma(X)$ are the vocabulary items (i.e., "types") associated with X and $w_U(u)$ indicates the undesirability of word u (see [28]). Minimizing f thus expresses a desire to have a small vocabulary.

4. Experiments and Results

Comparison of Different Algorithms: In the previous section we proposed three different algorithms with different $f(\cdot)$ and $g(\cdot)$ instantiations. Our goal here is to create subsets using different submodular optimization algorithms as well as function instantiations. For Switchboard I and Fisher we first remove utterances containing the disfluencies and fillers. For example, we remove utterances that contain only word fragments (e.g. sim[ilar]-), *uh*, [noise], *yeah*, [laughter], *huh*, *hm*, [laughter-*], *uh-huh*, *um-hum hum*, *huh-uh*, *um*. The size of the resulting ground sets V for Switchboard I and Fisher is 93312, and 1.7 million, respectively. For Switchboard I, we use a combination of different algorithms and function instantiations for a given target vocabulary size. For Fisher, we use the SCSK algorithm with g_5 as the quality function because of scalability and the computational efficiency of the SCSK algorithm.

To choose the best resulting corpus for a particular target vocabulary size, we run each of the optimization methods (Sections 3.1, 3.2, and 3.3) which gives us a relatively small number of corpora to choose from. We then compute a set of statistics on each of the resulting corpora such as the actual vocabulary size, average number of phonemes per word, the number of utterances and tokens, the speech durations, etc. We also compute the value g_5 for each subset; note that the g_5 function measures the representativeness of the subsets. We believe this value is a good indicator of corpus diversity. To show a corpus's phonetic balance, we compute the normalized entropy of the phoneme distribution $\frac{H(p)}{\log(43)}$ and the normalized entropy of the non-silence phoneme distribution $\frac{H(p)}{\log(42)}$, where we use 43 phones in the lexicon with 42 non-silence phones. $H(p)$ is the entropy of the probability distribution over phonemes in the selected subset.

In order to have the best final corpus for the current vocabulary size, we make the final selection by visual inspection of these statistics (i.e., by hand).[1] Table 1 shows the statistics of our chosen corpora, comprising both SVitchboard II and FiSVer I. Table 2 shows the specific algorithm and function instantiations we used to create each subset in Svitchboard II.

Task	Algorithm and Function
50	DS, g_5
100	SFM, g_2
500	DS, g_5
1000	DS, g_5
5000	SFM, g_2
10000	SFM, g_2

Table 2: Selected datasets for Svitchboard-II and the corresponding algorithms and functions.

[1] We will release all corpora that resulted from Sections 3.1, 3.2, and 3.3, although we here run baselines experiments only on our chosen sets.

Data Partition for Cross-Validation: For SVitchboard-II, our baselines define a cross-validation procedure. The conversation sides in each subsets are split into 5 non-overlapping folds; each conversation only exists in one fold. Similar to [10], we denote these five folds as sets A, B, C, D, and E (with no conversation side overlaps). For each vocabulary task, we create 5 subtasks: we use 4 out of the 5 folds as training data. For development data, we use the first half of the remaining fold; for evaluation data, we use the second half of the remaining fold. Table 3 shows the cross-validation schemes of SVitchboard-II. For the FiSVer-I 10-vocabulary and 50-vocabulary subsets, we split the first 90% utterances as training data, and the remaining 5% and 5% utterances as used as development and evaluation set, respectively. For other vocabulary sizes we split the data into 98%, 1% and 1% as training, development and evaluation sets, respectively. A trigram language model is built for each experiment. For each subtask in SVitchboard-II, the language models must be trained *only* on the training data as shown in Table 3, *not* on the entire data before the splitting.

Subtask	Train	Dev	Eval
1	$ABCD$	E_1	E_2
2	$BCDE$	A_1	A_2
3	$CDEA$	B_1	B_2
4	$DEAB$	C_1	C_2
5	$EABC$	D_1	D_2

Table 3: Five-fold cross-validation schemes of SVitchboard-II. A-E corresponds the five non-overlapping folds of the original dataset. The numbers in subscripts denote the first half or second half of the block.

Baseline Experiments: For each task, we establish two baseline systems, one with a triphone GMM-HMM system, and the other with a triphone DNN-HMM system, both of which are trained using the Kaldi open-source toolkit [31]. For the GMM-HMM system, we first flat-start a monophone GMM-HMM system, with 13 MFCCs and their deltas and delta-deltas (MFCC+Δ+$\Delta\Delta$). Cepstral mean normalization is performed for each conversation side. After the monophone system has been trained, we use it to train a context-dependent GMM triphone model with MFCC+Δ+$\Delta\Delta$ features. The total number of Gaussians for each task is around 25k. For the DNN-HMM system, we use alignments from the GMM-HMM system to bootstrap a triphone DNN/HMM system. Kaldi supports two different DNN training schemes: the first one is based on [32], which includes standard Restricted Boltzmann Machines (RBM), pre-training and stochastic gradient descent (SGD) training with GPUs; the second one supports parallel training on multiple CPUs and GPUs, and uses greedy layer-wise supervised training or layer-wise backpropagation [33, 34] and is described in detail in [35]. We use the second DNN training recipe with GPUs to create our baselines: for each task, we create a 4-layer network, with 1024 nodes in each network. The input features are spliced MFCCs (with a context window size of 4), followed by an LDA transformation (without dimensionality reduction) which is used to decorrelate the input features. The resulting feature vector has 117 dimensions in total. We use 20 epochs to train the DNN, with a mini-batch size of 256. For the first 15 epochs, we decrease the learning rate from 0.01 to 0.001 and fix the learning rate at 0.001 for the last 5 epochs. The number of Gaussian components is around 25k for each system. The numbers of parameters in the DNN systems are around 3.8 millions and 4.0 millions for the 50/100-vocabulary tasks, and around 5.2 millions for others. The baselines results for SVitchboard-II and FiSVer-I are shown in Table 4 and Table 5, respectively. We also run the same system on the 109-hour Switchboard and obtained a WER of 46.4% and 31.6% on the Hub-5 Eval 2000 dataset, which is comparable to previous work [36] with a similar setup.

Task	Subtask	GMM-HMM		DNN-HMM	
		Dev	Eval	Dev	Eval
50	1	35.37%	36.98%	26.89%	29.93%
	2	34.90%	31.97%	28.32%	26.78%
	3	32.39%	35.63%	27.61%	30.18%
	4	35.14%	31.29%	28.61%	25.45%
	5	32.05%	34.16%	26.34%	26.49%
100	1	39.13%	41.53%	29.74%	33.18%
	2	38.85%	36.31%	32.26%	30.74%
	3	36.05%	39.09%	31.06%	32.50%
	4	38.47%	35.11%	31.04%	27.98%
	5	35.51%	36.77%	28.68%	28.24%
500	1	44.85%	43.15%	37.86%	35.00%
	2	41.75%	40.74%	33.85%	32.72%
	3	43.18%	44.48%	35.64%	36.96%
	4	42.27%	40.96%	34.29%	33.00%
	5	42.58%	40.44%	33.62%	32.52%
1000	1	44.54%	43.77%	36.14%	34.21%
	2	42.11%	41.97%	33.22%	32.10%
	3	44.31%	46.75%	35.59%	37.08%
	4	42.89%	41.29%	33.18%	31.69%
	5	43.84%	40.99%	34.17%	32.17%
5000	1	44.52%	44.23%	33.69%	33.49%
	2	40.52%	40.68%	30.19%	30.34%
	3	45.10%	45.79%	35.74%	34.63%
	4	42.86%	40.61%	31.96%	29.30%
	5	43.53%	41.56%	32.84%	31.28%
10000	1	45.22%	45.26%	32.04%	32.17%
	2	41.28%	41.50%	29.01%	29.20%
	3	45.16%	46.74%	31.25%	33.14%
	4	43.03%	40.31%	30.05%	27.21%
	5	44.53%	43.00%	31.47%	30.43%

Table 4: Baseline results (word error rates) on SVitchboard-II using GMM-HMM systems and DNN-HMM systems.

Task	GMM-HMM		DNN-HMM	
	Dev	Eval	Dev	Eval
10	3.76%	8.55%	2.25%	5.43%
50	11.37%	15.69%	8.66%	13.30%
100	25.09%	20.71%	19.03%	17.89%
500	36.97%	32.69%	27.87%	23.80%
1000	41.91%	39.95%	31.14%	29.11%

Table 5: Baseline results (word error rates) on FiSVer-I using GMM-HMM systems and DNN-HMM systems.

5. Conclusions

We have introduce a new set of benchmark corpora derived from the Switchboard-I and Fisher datasets. Our goal is to provide to the ASR and machine learning communities high-quality limited-complexity corpora of conversational English speech. The resulting SVitchboard-II and FiSVer-I datasets will hopefully enable researchers to conduct experiments on novel machine learning algorithms and acoustic modeling methods without inordinate turnaround time. The data can be downloaded from `https://bitbucket.org/melodi/hqlc-speechcorpora`.

6. Acknowledgments

We thank the anonymous reviewers for their suggestions and comments. This material is based upon work supported by the National Science Foundation under Grant No. IIS-1162606, and by a Google, a Microsoft, and an Intel research award. Rishabh Iyer acknowledges support from the Microsoft Research Ph.D Fellowship.

7. References

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

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

[3] T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8614–8618.

[4] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6645–6649.

[5] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2014.

[6] H. Sak, O. Vinyals, G. Heigold, A. Senior, E. McDermott, R. Monga, and M. Mao, "Sequence discriminative distributed training of long short-term memory recurrent neural networks," 2014.

[7] G. Saon, H.-K. J. Kuo, S. Rennie, and M. Picheny, "The ibm 2015 english conversational telephone speech recognition system," *arXiv preprint arXiv:1505.05899*, 2015.

[8] J. Godfrey, E. Holliman, and J. McDaniel, "Switchboard: telephone speech corpus for research and development," in *Proceedings of ICASSP*, 1992, pp. 517–520.

[9] C. Cieri, D. Miller, and K. Walker, "The Fisher corpus: a resource for the next-generation speech-to-text," in *Proceedings of LREC*, 2004.

[10] S. King, C. Bartels, and J. Bilmes, "SVitchboard 1: Small Vocabulary Tasks from Switchboard," in *Ninth European Conference on Speech Communication and Technology*. ISCA, 2005.

[11] H. Lin and J. A. Bilmes, "Optimal selection of limited vocabulary speech corpora," in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, Florence, Italy, August 2011.

[12] S. Fujishige, *Submodular functions and optimization*. Elsevier Science Ltd, 2005.

[13] A. Krause, B. McMahan, C. Guestrin, and A. Gupta, "Robust submodular observation selection," *Journal of Machine Learning Research (JMLR)*, vol. 9, pp. 2761–2801, 2008.

[14] A. Das and D. Kempe, "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection," in *ICML*, 2011.

[15] Y. Liu, K. Wei, K. Kirchhoff, Y. Song, and J. Bilmes, "Submodular feature selection for high-dimensional acoustic score space," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2013.

[16] K. Kirchhoff, Y. Liu, and J. Bilmes, "Classification of developmental disorders from speech signals using submodular feature selection," in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, Lyon, France, August 2013.

[17] M. Narasimhan, N. Jojic, and J. Bilmes, "Q-clustering," in *NIPS*, 2005.

[18] M. Narasimhan and J. Bilmes, "PAC-learning bounded tree-width graphical models," in *Uncertainty in Artificial Intelligence: Proceedings of the Twentieth Conference (UAI-2004)*. Morgan Kaufmann Publishers, July 2004.

[19] H. Lin and J. Bilmes, "A class of submodular functions for document summarization," in *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, Portland, OR, June 2011.

[20] ——, "Learning mixtures of submodular shells with application to document summarization," in *Uncertainty in Artificial Intelligence (UAI)*. Catalina Island, USA: AUAI, July 2012.

[21] S. Tschiatschek, R. Iyer, H. Wei, and J. Bilmes, "Learning mixtures of submodular functions for image collection summarization," in *Neural Information Processing Society (NIPS)*, Montreal, CA, December 2014.

[22] H. Lin and J. A. Bilmes, "How to select a good training-data subset for transcription: Submodular active selection for sequences," in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, Brighton, UK, September 2009.

[23] K. Wei, Y. Liu, K. Kirchhoff, C. Bartels, and J. Bilmes, "Submodular subset selection for large-scale speech training data," in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, Florence, Italy, 2014.

[24] A. Krause, J. Leskovec, C. Guestrin, J. VanBriesen, and C. Faloutsos, "Efficient sensor placement optimization for securing large water distribution networks," *Journal of Water Resources Planning and Management*, vol. 134, no. 6, pp. 516–526, 2008.

[25] R. Iyer and J. Bilmes, "Submodular optimization with submodular cover and submodular knapsack constraints," in *Neural Information Processing Society (NIPS)*, Lake Tahoe, CA, December 2013.

[26] ——, "Algorithms for approximate minimization of the difference between submodular functions, with applications," *In UAI*, 2012.

[27] M. Narasimhan and J. Bilmes, "A submodular-supermodular procedure with applications to discriminative structure learning," in *UAI*, 2005.

[28] H. Lin and J. Bilmes, "An application of the submodular principal partition to training data subset selection," in *NIPS Workshop on Discrete Optimization in Machine Learning: Submodularity, Sparsity & Polyhedra*, Vancouver, Canada, December 2010.

[29] K. Kirchhoff and J. Bilmes, "Submodularity for data selection in machine translation," in *Empirical Methods in Natural Language Processing (EMNLP)*, October 2014.

[30] P. Stobbe and A. Krause, "Efficient minimization of decomposable submodular functions," in *NIPS*, 2010.

[31] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011, pp. 1–4.

[32] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. INTERSPEECH*, 2013, pp. 2345–2349.

[33] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle *et al.*, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, p. 153, 2007.

[34] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Interspeech*, 2011, pp. 437–440.

[35] D. Povey, X. Zhang, and S. Khudanpur, "Parallel training of deep neural networks with natural gradient and parameter averaging," *arXiv preprint arXiv:1410.7455*, 2014.

[36] L. Lu and S. Renals, "Probabilistic linear discriminant analysis with bottleneck features for speech recognition," in *Proc. INTERSPEECH*, 2014.