Language ID-based training of multilingual stacked bottleneck features

Yu Zhang, Ekapol Chuangsuwanich, James Glass

MIT Computer Science and Artificial Intelligence Laboratory,
Cambridge, Massachusetts 02139, USA

{yzhang87, ekapolc, glass}@mit.edu

Abstract

In this paper, we explore multilingual feature-level data sharing via Deep Neural Network (DNN) stacked bottleneck features. Given a set of available source languages, we apply language identification to pick the language most similar to the target language, for more efficient use of multilingual resources. Our experiments with IARPA-Babel languages show that bottleneck features trained on the most similar source language perform better than those trained on all available source languages. Further analysis suggests that only data similar to the target language is useful for multilingual training.

Index Terms: Multilingual, Bottleneck features, DNN

1. Introduction

Developing an automatic speech recognition (ASR) capability for a new language requires significant linguistic resources in the form of annotated data for acoustic and language modeling, and pronunciation dictionaries. These resources can be expensive and time-consuming to produce, and, as a result, have greatly limited the number of languages that currently have ASR capability. They are also a significant impediment to the rapid development of ASR capability for a new language. Moreover, given that ASRs perform best with substantial amounts of data, a related challenge is to obtain good performance when limited linguistic resources are available.

To mitigate the problem of limited data, researchers have explored the use of bottleneck (BN) features derived from Deep Neural Networks (DNNs). Recent progress on DNN-based acoustic modeling has greatly improved ASR performance on many tasks [1]. One way to apply DNNs in ASR is via BN features in a *tandem* approach [2, 3, 4, 5]. In this approach, a standard DNN with one smaller hidden layer, called the bottleneck layer, is trained. Then, the outputs of the bottleneck layer are used in conjunction with other features to train a standard GMM-HMM recognizer. Recent work (e.g., [6]) has shown that BN features achieve competitive results on IARPA Babel tasks.

Researchers have also used BN features to leverage out-of-domain resources, that are either multilingual [7, 8], or cross-lingual [9]. With access to larger amounts of data, BN features are able to better learn the structure of speech, and improve the performance on a target task. In [10], the target language data is also used for adaptation of the multilingual DNN by doing additional fine-tuning steps. These approaches not only alleviate the lack of training data, they also save the amount of time required to train DNNs for the target languages.

None of the work mentioned above addresses the issue of what to do when there are multiple source BN systems to choose from, i.e. having one BN system for each language. This is not an unrealistic scenario, as researchers often have multiple recognizers on hand. Furthermore, multilingual DNNs require modifications of the existing training strategies, and take longer to train. In this work, we propose a simple Language Identification (LID) method to select possible candidate languages for transfer learning. Experiments show that BN systems trained on languages close to the target language can yield better performance than BN systems trained in a multilingual fashion. We also employ a two-stage training strategy where the selected source language is used in conjunction with the multilingual DNN to improve the performance even further.

The rest of the paper is organized as follows. In Section 2 we review the BN architecture used in our previous work. In Section 3 we describe the Babel corpus and the goals of the Babel project. In Section 4, we demonstrate the potential of data selection by LID. Then, we explore different possible training strategies in Section 5. We also offer insight into why language selection or data selection in general is important for cross domain adaptation. Finally, we conclude the paper in Section 6.

2. Stacked Bottleneck Architecture

The BN features used in this work follow our previous work in [6]. As shown in Figure 1, our BN extraction is a concatenation of two DNNs. The outputs from the BN layer in the first DNN are used as the input features for the second DNN, whose outputs at the BN layer are then used as the final features for standard GMM-HMM training. Unlike most research with DNN-based BN features, our BN layer uses a linear activation function to enforce a low-rank approximation of the softmax layers.

The inputs of the first layer consist of 23 critical-band energies obtained from a Mel filter-bank. Each of the 23 dimensions are augmented with pitch and probability of voicing [11] and multiplied across time by a Hamming window of length 11 frames. A DCT is then applied for dimensionality reduction. The 0^{th} to 5^{th} coefficients are retained, resulting in a feature of dimensionality $(23 + 2) * 6 = 150$. The input features of the second DNN are the outputs of the BN layer from the first DNN. Context expansion is done by concatenating frames with time offsets $-10, -5, 0, 5, 10$. Thus, the overall time context seen by the second DNN is 31 frames. Both DNNs use same setup of 5 hidden sigmoid layers (1024 hidden units) and 1 linear BN layer (80 hidden units). Both of them use tied context-

Supported in part by the Intelligence Advanced Research Projects Activity(IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0013. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government

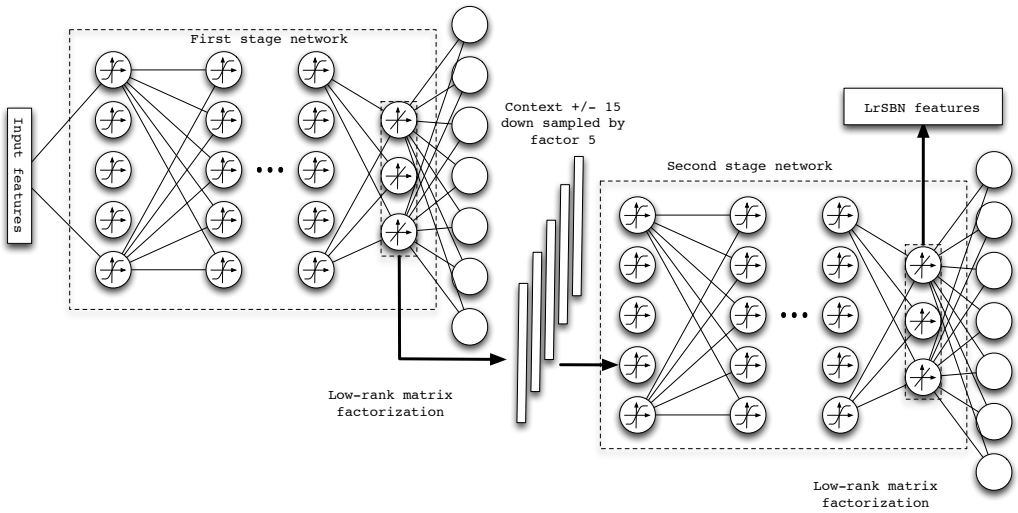

Figure 1: Diagram of the stacked bottleneck neural network feature extraction framework.

dependent (CD) states as target outputs, which are generated by forced alignment from a GMM-HMM baseline. No pre-training is used. A final PCA is applied on the second set of BN outputs to reduce the dimension to 30. Lastly, delta and delta-delta features are concatenated resulting in a final dimensionality of 90.

3. Babel corpus

The IARPA Babel program focuses on ASR and spoken term detection on low-resource languages [12]. The goal of the program is to reduce the amount of time needed to develop an ASR and spoken term detection capability in a new language. The rapid development aspect puts constraints on both the amount of in-domain language resources collected, such as transcribed speech from the language, and the training time (decreasing from one month to one week over the course of the program). Every year the program intends to release 5-6 new low-resource languages, meaning that by the end of the program there will be more than 20 languages available. With each passing year we will have access to more recognizers that are already trained for the languages from the previous years, so methods that can make use of previous systems are desirable. In this work, we put emphasis on the training and deployment conditions that would make sense under the Babel program. However, we believe that many of these constraints make sense for other scenarios as well.

The data from the Babel project consists of collections of speech, both transcribed and un-transcribed, from a growing list of languages. Currently the project is on its second year. The languages from the first year include Cantonese, Turkish, Pashto, Tagalog and Vietnamese. The second year languages are Bengali, Assamese, Zulu, Lao, and Haitian Creole. Each language consists of two speaking styles: scripted (prompted speech) and conversational (spontaneous telephone conversations). There are also currently two standard training conditions: Full (∼80 hours of transcribed speech) and Limited (10 hours of transcribed speech). Most data are recorded via landline or cell-phone, while Lao, Haitian, and Zulu also include a small amount of wideband data recorded by microphones. For this work, we downsample the wideband data to 8kHz, and process all data equivalently. We focus on the Limited condition for target languages, while the source languages have access to the Full condition. In our experiments, we only use the conversational data for training. The standard 10-hour dev sets containing only conversational data are used for testing.

4. Language pair transfer learning

In this section, we motivate the potential benefits of language pair transfer learning, and whether these systems, especially the trained DNNs, can be used to facilitate the training of new target languages.

4.1. A case study on Assamese and Bengali

We start by looking at the best case scenario possible, namely the language pair of Assamese and Bengali. Assamese and Bengali are spoken in adjacent regions in India. They are known to be linguistically close, with overlapping phoneme inventories and vocabulary. We used Limited Bengali (IARPA-babel102b-v0.4) and Full Assamese (IARPA-babel103b-v0.3) as the target and source languages, respectively. The transfer learning is done by using the BN features trained on Full Assamese to extract BN features for Limited Bengali. Tied-state triphone CD-HMMs, with 2500 states, and 18 Gaussian components per state, were used for acoustic modeling. Discriminative training was done using the Minimum Bayes risk (MBR) criterion [13]. For language modeling, a trigram LM is learned from training data transcripts. We use the same recognizer setup for the rest of the paper.

As shown in Table 1, using the Assamese BN features improved the WER by 1.4% absolute over the Limited Bengali BN baseline system. We can also perform additional adaptation on the DNNs from Assamese. This is done by replacing the original softmax layer with a randomly initialized Bengali softmax layer, and performing additional fine-tuning iterations. Replacing the softmax layer completely eliminates the need to do phoneme mapping between languages. This adaptation process is equivalent to using the Assamese data to "pre-train" the Bengali network, which helps initialize the DNNs into a better starting point. With the better initialization the network typically converges in 5 iterations instead of the 10 iterations needed for a randomly initialized network. The adaptation is done on both DNNs, which reduces the WER even further to 63.7%. As an Oracle baseline, we also use a BN system trained on Full

System	WER (%)
Limited PLP	71.8
Full PLP	64.5
Limited BN	66.0
Full BN	55.4
Limited + Full Assamese BN	64.6
Limited + Adapted Full Assamese BN	63.7
Limited + Full Bengali BN	61.6

Table 1: WER on Bengali with different data usage scenarios.

Target (Limited)	Source BN (Full)			
	Bengali	Assamese	Lao	Turkish
Bengali (66.0)		**63.7**	65.1	64.2
Assamese (65.2)	**61.2**		62.9	62.1
Lao (62.3)	**59.8**	60.1		60.0
Turkish (63.9)	**61.8**	63.1	63.3	

Table 2: WER using between different language pairs. Numbers in parenthesis are Limited Monolingual BN baseline.

Bengali to extract features for the Limited Bengali system. The WER of this setup is 61.6%. Thus, the adapted Assamese BN system is able to capture 52% of the gain that would be achieved by using more supervised data on Bengali to train the DNNs.

4.2. Other language pairs

To look at the possibility of transfer learning in a broader scenario, where the languages are less similar, we expand our experiments to include two more languages, namely Lao (IARPA-babel203b-v2.1a) and Turkish (IARPA-babel105b-v0.4). Table 2 summarizes the BN feature transfer learning WERs with target language adaptation on the four languages. As expected, the closest language pair of Assamese and Bengali seems to mutually benefit the most. Bengali also seems to be a good language in general for the other three languages.

4.3. Language ID for source language selection

Although the experiments in Section 4.2 are promising, in most cases language similarities are far from obvious, and the prospect of trying out all possible source languages might not be time efficient. We propose to use Language Identification (LID) as a way to determine which language to use as the source language. We start by training a DNN with 2 hidden layers and 512 hidden units per layer for LID on the four languages. We randomly selected 90% of the Limited training sets for training the network. Unlike the DNN-based LID work in [14], we use the same input features as the ones described in Section 2. This is to make the LID DNN decide which languages are similar based on what the BN DNN would observe. We then use the DNN to classify the remaining 10% of the (held out) data. Table 3 summarizes the posteriors of each language, averaged across all frames. The closeness between Assamese and Bengali are again confirmed by the LID results, with average posteriors of 0.21. Turkish is also closest to Bengali, which is consistent with our previous experiment. Less similar pairs seem to also correspond to worse WERs in the previous experiment. The only language that does not follow the predicted trend seems to be Lao. However, the WER difference between using the closest language (in the LID sense) and the best possible outcome is only 0.3%. Thus, we believe that LID is a reasonable method to select a source language for transfer learning.

Input frames	Predicted posteriors (Averaged)			
	Bengali	Assamese	Lao	Turkish
Bengali	0.57	**0.21**	0.09	0.13
Assamese	**0.21**	0.57	0.11	0.11
Lao	0.08	**0.11**	0.71	0.10
Turkish	**0.13**	0.12	0.10	0.65

Table 3: Average posteriors for the initial LID experiment.

5. Multilingual Experiments

In this section, we compare different multilingual strategies and their training time trade-off. For a stronger baseline, we modified the BN features described in Section 2 as follows. The filterbank inputs were processed with VTLN warping factors [15]. Kaldi's pitch extractor [16] and Fundamental Frequency Variation (FFV) features [17] are used instead of Subband Autocorrelation Classification pitch tracker (SAcC) [11]. Speaker adaptation is also applied to the outputs of the first BN DNN before feeding it to the second BN DNN [15]. Only year one languages, namely Cantonese (IARPA-babel101-v0.4c), Turkish (IARPA-babel105b-v0.4), Pashto (IARPA-babel104b-v0.4aY), Tagalog (IARPA-babel106-v0.2g) and Vietnamese (IARPA-babel107b-v0.7), are considered as source languages.

A multilingual stacked bottleneck DNN is trained on the Full condition of all year one languages which consists of $\sim$300 hours. The DNN training follows [18] where all the DNN targets from each language are pooled together. This has an effect of doing discrimination against all targets of the other language as well[1]. Language-specific speaker adaptation then is applied on the outputs of the first DNN. Similarly, the monolingual versions of all the year one languages are trained using this procedure.

5.1. Adaptation strategies

Since there are two DNNs in the stacked BN architecture, several adaptation strategies are available. In [10], they observed that it is beneficial to adapt the first DNN. However, the second DNN should be either adapted, or trained from scratch using the target data only, depending on the target language. Thus, for our experiments, we always adapt the first DNN from either the multilingual DNN or the monolingual DNNs from year one. For the second DNN, the following approaches were considered:

1. Training from scratch using only the target language data (target only).
2. Adapting from the multilingual DNN (multi).
3. Adapting from a monolingual DNN from year one (mono).
4. Re-train the monolingual DNN using the features from the first multilingual DNN which is already adapted to the target language. Then, use the target language data to do adaptation (mono re-train).
5. Re-train the multilingual DNN using the features from the first multilingual DNN which is already adapted to the target language. Then, use the target language data to do adaptation (multi re-train).

[1]From the work in [10], there was a small difference in WER between this approach and the one where the targets of each language are discriminated only amongst themselves. We chose the current approach due to ease of implementation.

Input frames	Predicted posteriors (Averaged)				
	T	P	U	C	V
Lao	0.25	0.08	0.14	0.16	**0.37**
Assamese	**0.31**	0.18	0.19	0.06	0.26

Table 4: LID posteriors of the year one languages; Tagalog (T), Pashto (P), Turkish (U), Cantonese (C), and Vietnamese (V).

	DNN for adaptation		WER (%)	
	1^{st} stage	2^{nd} stage	Lao	Assamese
a	target only	target only	61.5	63.3
b	multi	target only	59.0	61.2
c	multi	multi	57.5	59.4
d	multi	mono	58.0 (P)	60.1 (P)
e	multi	LID	57.5 (V)	59.4 (T)
f	LID	LID	56.8 (V)	59.3 (T)
g	LID	LID re-train	56.5 (V)	59.0 (T)
h	multi	LID re-train	56.0 (V)	58.5 (T)
i	multi	mono re-train	56.7 (P)	58.8 (P)

Table 5: WER on Limited Lao and Limited Assamese using different adaptation strategies. Letters in parentheses denote the source language used for the monolingual DNNs.

While the original multilingual DNN can be trained in advance, we do not consider method 5 since re-training the multilingual DNN would take longer than one week. Re-training a Full condition's worth of data (method 4) would take around 13 hours. The adaptation of the DNNs with 10 hours of target language data (all methods) can be done in less than one hour.

5.2. Results

For our multilingual experiments we chose Limited Lao and Limited Assamese as our target languages. A DNN for LID was trained for the five year one languages as described in Section 4.3. The average predicted posteriors from the Limited condition of the two target languages are summarized in Table 4. Lao is closest to Vietnamese, while Assamese is closest to Tagalog. Note that Pashto and Assamese fall under the same language family of Indo-Iranian, but Pashto is placed fourth in terms of LID similarity to Assamese.

Table 5 shows the results of the different adaptation strategies where LID denotes using the monolingual DNN from the closest year one language. For comparison, we also use Pastho as another possible source language. The baseline BN systems using only the target language data have a WER of 61.5% and 63.3% for Lao and Assamese, respectively. All the multilingual systems perform better than the baseline, showing the benefits of using additional resources to facilitate low-resource ASR. Adapting both DNNs improves the WER in all cases. As expected, using the closest language DNN performs better than Pashto (d vs. e). More importantly, using the monolingual DNN from the closest languages for the first and second DNNs works slightly better than the multilingual counterparts (f vs. c). The multilingual DNN seems to help when coupled with re-training of the second DNN using the closest language (h), improving the WER by another 0.8% for both Lao and Assamese. This, however, comes with a slightly longer training time.

5.3. No data like similar data

The experiments in the previous section show that the language identified as the closest identified language alone can achieve comparable performance to the combined multilingual training. Yet, the data used in the monolingual systems are strictly subsets of the multilingual data. This seems to imply that including other languages which are "further away" can hurt performance. To this end, we re-visit the Lao-Turkish (source-target) language pair, which provided the least performance gain in Section 4.2.

Amount	Lao data usage	WER (%)
0	None (Turkish only)	63.9
10hrs	Limited condition data	64.4
10hrs	Furthest utterances	66.5
10hrs	Closest utterances	63.8
10hrs	Closest frames	63.1
65 hrs	Full condition data	63.3
32.5 hrs	Random utterances	64.0
32.5 hrs	Closest utterances	62.8
32.5 hrs	Closest frames	62.4

Table 6: Effect of source data selection on Limited Turkish.

In the same way that LID can identify which language is closest to the target data, LID can also be used as a selection tool to determine which portion of the data is most useful for the target language. Suppose we train a LID DNN using all the data from the Limited condition of Lao and Turkish. Then, for BN DNN training, we select Lao data at either the frame or utterance level. At the utterance level, the frame posteriors are averaged across each utterance. The frames/utterances with highest posteriors are then used to train the source BN DNN for further adaptation.

Table 6 summarizes the different data selection strategies. Using the provided Limited Lao subset, the performance is even worse than the baseline with no Lao data at all. Unsurprisingly, the performance degrades even further if 10 hours of the furthest utterances from Full Lao are used. Using 10 hours of the closest utterances, however, can achieve a WER of 63.8%, slightly better than the Limited Turkish baseline. Selecting based on frames gives a slightly better WER than utterance-based selection. The best performing system based on 10 hours of Lao data selects only the closest frames and attains an WER of 63.1%. With only one-sixth of the data, we do as well as if we had used the Full Lao. Selecting the closest half of the frames yields a WER of 62.4%, a 0.9% absolute improvement.

From the results, having data that is similar to the target data seems to be more important than having more source data. This anecdotal observation seems to suggest that adequately robust BN features can be trained without much data, especially when the resulting DNNs are used as a starting point for adaptation. As less similar data would put the DNNs into worse initializations, perhaps we should exercise more care in selecting data for multilingual adaptation.

6. Conclusion

We investigated the use of LID for language selection to facilitate multilingual training in a framework for extracting stacked BN features. Experiments showed that monolingual DNNs from "close" languages could outperform a full multilingual training, with the combination of the two yielding the best results. For future work, we plan to explore data selection to select frames across multiple languages and doing speaker adaptation via i-vectors instead of explicit feature transforms [19, 20].

7. References

[1] G. Hinton, L. Deng, D. Yu *et al.*, "Deep neural networks for acoustic modeling in speech recognition," in *IEEE Signal Processing Magazine*, vol. 28, no. 6, November 2012, pp. 82–97.

[2] D. Yu and M. L. Seltzer, "Improved bottleneck features using pretrained deep neural networks," in *Proc. InterSpeech*, 2011, pp. 273–240.

[3] T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *Proc. ICASSP*, 2012, pp. 4153–4156.

[4] Z. J. Yan, Q. Huo, and J. Xu, "A scalable approach to using DNN-derived features in GMM-HMM based acoustic modeling for lvcsr," in *Proc. InterSpeech*, 2013.

[5] K. Veselý, M. Karafiát, and F. Grézl, "Convolutive bottleneck network features for LVCSR," in *Proc. ASRU*, 2011, pp. 42–47.

[6] Y. Zhang, E. Chuangsuwanich, and J. Glass, "Extracting deep neural network bottleneck features using low-rank matrix factorization," in *Proc. ICASSP*, 2014.

[7] K. Veselý, M. Karafiát, F. Grézl *et al.*, "The language-independent bottlneck features," in *Proc. SLT*, 2012.

[8] N. T. Vu, F. Metze, and T. Schultz, "Multilingual bottle-neck features and its application for under-resourced languages," in *Proc. SLT*, 2012.

[9] A. Stolcke, F. Grézl, M. Hwang *et al.*, "Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons," in *Proc. ICASSP*, 2006.

[10] F. Grézl, M. Karafiát, and K. Veselý, "Adaptation of multilingual stacked bottle-neck neural network structure for new languages," in *Proc. ICASSP*, 2014.

[11] D. Ellis and B. Lee, "Noise robust pitch tracking by subband autocorrelation classication," in *13th Annual Conference of the International Speech Communication Association*, 2012.

[12] *IARPA broad agency announcement IARPA-BAA-11-02*, 2011.

[13] M. Gibson and T. Hain, "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition," in *Proc. InterSpeech*, 2006, pp. 2406–2409.

[14] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, and D. Martinez, "Automatic language identification using deep neural networks," in *Proc. ICASSP*, 2014.

[15] T. N. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for lvcsr," in *Proc. ICASSP*, 2013.

[16] P. Ghahremani, B. BabaAli, K. R. D. Povey, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," 2014.

[17] K. Laskowski, M. Heldner, and J. Edlund, "The fundamental frequency variation spectrum," in *Proc. FONETIK*, 2008.

[18] F. Grézl, M. Karafiát, and M. Janda, "Study of probabilistic and bottle-neck features in multilingual environment," in *Proc. ASRU*, 2011.

[19] G. Saon, H. Soltau, D. Hanamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proc. ASRU*, 2013.

[20] A. Senior and I. L. Moreno, "Improving DNN speaker independence with i-vector inputs," in *Proc. ICASSP*, 2014.