TELKOMNIKA, Vol.16, No.3, June 2018, pp. 1226~1238
ISSN: 1693-6930, accredited **A** by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v16i3.7984

Shared-hidden-layer Deep Neural Network for Under-resourced Language the Content

Devin Hoesen[1], Dessi Puji Lestari[2], Dwi Hendratmo Widyantoro[3]
Department of Informatics, Institut Teknologi Bandung,
Jl. Ganeca No. 10 Bandung, +62-22-2508135, Indonesia
*Corresponding author, e-mail: 23514103@std.stei.itb.ac.id[1], dessipuji@stei.itb.ac.id[2], dwi@stei.itb.ac.id[3]

Abstract

Training speech recognizer with under-resourced language data still proves difficult. Indonesian language is considered under-resourced because the lack of a standard speech corpus, text corpus, and dictionary. In this research, the efficacy of augmenting limited Indonesian speech training data with highly-resourced-language training data, such as English, to train Indonesian speech recognizer was analyzed. The training was performed in form of shared-hidden-layer deep-neural-network (SHL-DNN) training. An SHL-DNN has language-independent hidden layers and can be pre-trained and trained using multilingual training data without any difference with a monolingual deep neural network. The SHL-DNN using Indonesian and English speech training data proved effective for decreasing word error rate (WER) in decoding Indonesian dictated-speech by achieving 3.82% absolute decrease compared to a monolingual Indonesian hidden Markov model using Gaussian mixture model emission (GMM-HMM). The case was confirmed when the SHL-DNN was also employed to decode Indonesian spontaneous-speech by achieving 4.19% absolute WER decrease.

Keywords*: deep neural network, grapheme-to-phoneme, indonesian, shared hidden layer, under-resourced*

1. Introduction

Automatic Speech Recognition (ASR) training for under-resourced languages still proves to be a big challenge in speech recognition community [1-2]. Speech-recognition training is generally benefited from large training data because it heavily relies on the statistical distribution of speech features presented in them. Given limited training data, the speech parameter estimation involved in acoustic modeling may be compromised [3]. On the other hand, annotating large training data is difficult and time-consuming. As a result, many researches still struggle to find an effective method to train an ASR in such condition.

Indonesian language or Bahasa Indonesia is one of many under-resourced languages in terms of speech training. Despite having massive number of speakers, it still lacks a standard speech corpus, text corpus, and dictionary [4-5]. Therefore many researches have to develop their own corpus and/or dictionary.

Incorporating other language(s) to enrich the training data for recognizing the under-resourced language becomes an option. One method to achieve this is using a hidden Markov model with Gaussian mixture model emissions (GMM-HMM) based system in form of subspace Gaussian mixture model (SGMM). In [6], the system is trained with small Afrikaans training data incorporated with large Dutch data to recognize Afrikaans speech. Reference [7] also uses SGMM-based recognizer trained with German, Spanish, and limited English data to recognize English words. In both researches, there is improvement in recognition accuracy for the target language (Afrikaans and English respectively). However, it has been analyzed that Afrikaans is closely related to Dutch [8] and English belongs to the same language group as German and Spanish, i.e. Indo-European Group [9].

For Indonesian, there is no research that employed SGMM as the speech model. One research [10] incorporates English into Indonesian training data for the speech recognizer in a conventional GMM-HMM system. However, the recognition accuracy for Indonesian is degraded. Indonesian, the under-resourced language, is less related to English than Afrikaans

Received November 16, 2017; Revised April 1, 2018; Accepted April 27, 2018

to Dutch or English to German and Spanish. In regard to this result, there is still no research that analyzes the impact of SGMM for distantly related languages.

Meanwhile, many researches now analyze deep neural network (DNN) which has been state-of-the-art for speech recognition and can outperform GMM-HMM system(s) in most cases [11]. The DNN for speech recognition has more than one hidden layer and exactly one output layer. The DNN can utilize training data from many languages easily, as the hidden layers can be perceived as language-independent and only the output layer is language-dependent [3]. Moreover, DNN should be trained with large amount of data to reduce over-fitting [12]. Thus, a DNN can be effectively employed for recognizing under-resourced language by training it with the target language (the under-resourced language) training data incorporated with one or more highly-resourced language(s).

More researches also show that using the multilingual training for a DNN can benefit the under-resourced target language even if the non-target language(s) is/are distantly related with the target language. This is the case with a DNN trained with French, German, Spanish, and Italian training data. The hidden layers from the DNN are then transferred to an output layer to recognize Mandarin Chinese, a language that is distantly related to the four European languages. The DNN is called Shared-Hidden-Layer DNN (SHL-DNN). The SHL-DNN is then shown to outperform the monolingual DNN trained with only Mandarin Chinese by almost 10% absolute error reduction [13]. The case is also reconfirmed when an SHL-DNN is trained with English and limited amount of Indonesian data to recognize Indonesian speech. The SHL-DNN outperforms both DNN and GMM-HMM recognizer trained with only Indonesian data by almost 4% and 2% absolute error reduction respectively [14].

Another case to be researched is recognizing spontaneous speech. Spontaneous speech is different from dictated speech. Acoustically, the whole phoneme spectrums in spontaneous speech are more convergent while each of the phonemes has spectrums that are more diverse [15]. This makes inter-phoneme boundary diffuser. Linguistically, spontaneous speech usually contains filled pauses, repetitions, interjections, unknown or mispronounced words, omissions of pronouns and/or relatives, and ungrammatical sentences or unusual word orders [16]. These make recognizing spontaneous speech more challenging than dictated speech. While training a recognizer for an under-resourced language is challenging, training the recognizer to recognize spontaneous speech makes it more challenging. This is the case for Indonesian language where its spontaneous speech differs significantly from its dictated speech.

In this research, we show that using an SHL-DNN trained with the under-resourced Indonesian incorporated with the highly-resourced English training data can improve the Indonesian recognition accuracy. We also show that using the same SHL-DNN can improve the recognition accuracy for Indonesian spontaneous speech. Both cases are compared to a GMM-HMM system trained with only the under-resourced Indonesian data. We choose an SHL-DNN over an SGMM system due to the fact that DNN is the current state-of-the-art speech-recognition method.

2. Shared-hidden-layer Deep Neural Network (SHL-DNN)

2.1. Deep-neural-network hidden-markov-model (DNN-HMM)

Deep Neural Network (DNN), which is the state-of-the-art speech-recognition model, comprises many layers of hidden units and one output layer [11]. The hidden unit typically employs a sigmoid function, e.g. logistic function or sometimes hyperbolic tangent function. Each hidden unit j has an activation function, e.g. logistic function y_j which map its total input x_j into a scalar state and is defined as

$$y_j = \text{logistic}(x_j) = \frac{1}{1+\exp(-x_j)} \quad (1)$$

The total input xj for hidden layer j in layer i maps the output of each hidden units in the layer below (i - 1) into a value and is defined as

$$x_j = b_j + \sum_i w_{ij} y_i \qquad (2)$$

where b_j is a bias value for unit j and w_{ij} is a weight assigned to each connection from a hidden unit in layer (i-1) to unit j. For speech recognition, which is a multiclass classification, each hidden unit in the output layer k maps its total input to a class probability using "softmax" function, which is defined as

$$p_j = \frac{\exp(x^j)}{\sum_k \exp(x^k)}. \qquad (3)$$

Architecture of a fully-connected DNN is illustrated in Figure 1.

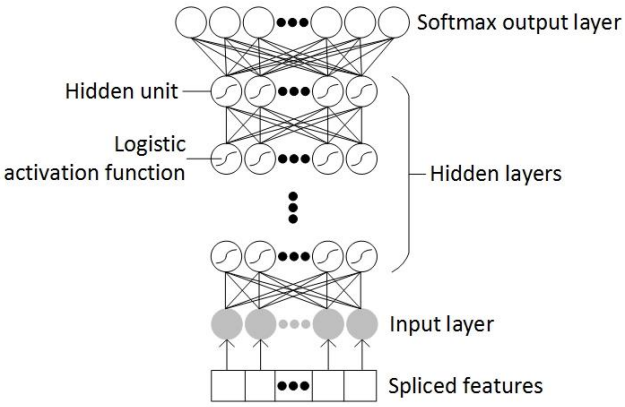

Figure 1. Architecture of a fully-connected DNN

A DNN can be discriminatively trained via back-propagation and gradient descent that measure discrepancies between the target outputs and the actual DNN outputs [17]. However, infinite variations of number of hidden layers and number of units in each hidden layer cause massive number of parameters to be predicted. Thus, it is difficult to optimize the model parameters [11]. A DNN with many hidden layers is a flexible model that is suitable to model complex data such as speech. However, it is difficult to find the best set of weights using the gradient descent algorithm that is initialized using a random point near the origin, unless the initial starting point is carefully chosen [18]. The weight initialization values will affect the back-propagated errors that will also affect each set of weights in each different layer.

One solution to this problem is to first generatively pre-train the DNN. Instead of randomly initializing the DNN's many hidden layers, a Deep Believe Network (DBN) can be constructed [19]. Each pair of adjacent layers in a DBN is constructed as a restricted Boltzmann machine (RBM). An RBM has one "visible" input layer and one "hidden" layer where each unit in the input layer is connected to each unit in the hidden layer but no unit has any connection to each other in the same layer [20]. By using an RBM, the first layer in a DBN is a visible input layer that transforms its input to an output that is fed to the second (hidden) layer. After the pair is constructed (trained), outputs of the second layer will act as inputs to a to-be-constructed third (hidden) layer. The training proceeds until the desired number of layers is achieved. By using the method, instead of assuming the DBN is good at discriminating classes, it is assumed to be good at modeling structure of the training data [11]. After the DBN is constructed, it can be employed as a good starting point to the gradient descent algorithm where back-propagated errors can slightly adjust its weights so that every unit is fine-tuned through a DNN training [21].

A DNN for speech recognition is interfaced with an HMM to become a hybrid DNN-HMM acoustic model. To compute the Viterbi alignment in an HMM system, scaled probabilities (likelihoods), $p(X|H)$, are required. They define probability of emitting feature X given an HMM

state *H*. However, a DNN gives posterior probabilitas, $p(H|X)$, which is probability of an HMM state *H* given a feature *X*. To convert the posterior probability into the scaled likelihood, consider the naïve Bayes rule,

$$p(X|H) = \frac{p(H|X)p(X)}{p(H)}. \quad (4)$$

Probability of an HMM state, $p(H)$, can be obtained from frequencies of the state in the aligned training data used to fine-tune the DNN. The probability of a feature, $p(X)$, is an unknown factor that has little effect on the alignment, so it can be ignored. While the conversion works in some recognition tasks where the data are balanced, it should be used with caution when the data are highly imbalanced (e.g. have many frames of silence) [11].

It should be noted that a DNN-HMM acoustic model can be trained with more-correlated features, unlike a GMM-HMM model that should be trained with highly decorrelated (independent) features. In a GMM-HMM model, if the features are strongly correlated, the HMM requires to utilize full covariance GMMs or larger number of diagonal GMMs and they make the computation more expensive. Hence, mel-frequency cepstral coefficients (MFCCs) [22], which are highly decorrelated between their individual feature components, are more suitable for GMM-HMM modeling [11]. On the other hand, a DNN training does not require highly decorrelated features, so we can use mel filterbanks [22], which are highly correlated between their individual feature components. In fact, a DBN-DNN trained with mel filterbanks achieves 1.7% lower absolute phone error rate than the best one trained with MFCCs [11],[23].

2.2. Shared-Hidden-Layer Deep-Neural-Network (SHL-DNN)

Shared-hidden-layer deep-neural-network (SHL-DNN) is a DNN trained with more than one language [13]. The input and hidden layers are shared across the languages while its output (softmax) layer is language-specific. Architecture of an SHL-DNN is illustrated in Figure 2.

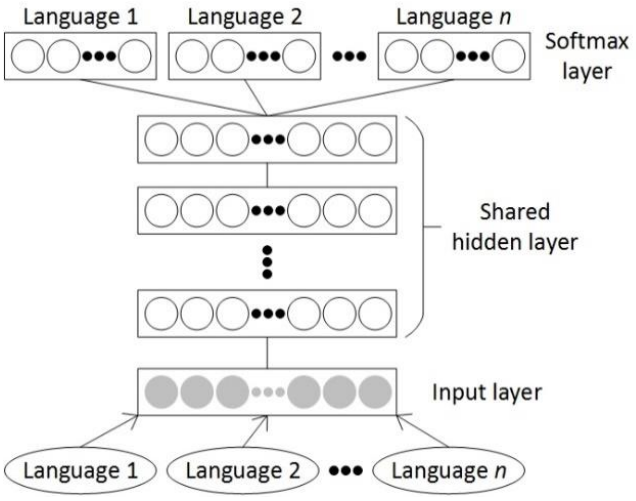

Figure 2. Architecture of an SHL-DNN

The input and shared hidden layers must be trained (or pre-trained) simultaneously with all decodable languages [13]. Fortunately, they can be pre-trained with the unsupervised DBN pre-training [19] since it does not require a language-specific output layer, so the pre-training can be performed rather easily [13]. It should also be noted that for each language to uniformly update the weight of all hidden units, the multilingual pre-training data must be shuffled uniformly across the languages.

After they have been pre-trained, they are fine-tuned with the usual back-propagation algorithm through a DNN training. However, the method to train an SHL-DNN must be slightly updated. An ordinary DNN has one softmax layer for one language, so discrepancies between its actual and its target outputs update all of its hidden and softmax units. In an SHL-DNN, each language has its own softmax layer, so discrepancies between its actual and its target outputs for a language update its hidden units and only the softmax units for the respective language.

An SHL-DNN trained with many languages can outperform a DNN trained with only one target language when recognizing the language. It is also shown that the shared hidden layers can be transferred to construct another DNN to recognize a target language that is distantly related with languages used to train the SHL-DNN. This is the case when an SHL-DNN trained with French, German, Spanish, and Italian to recognize Mandarin Chinese outperforms a monolingual DNN trained only in Mandarin by almost 10% absolute character error reduction [13]. This trait can benefit the under-resourced language, such as Indonesian, by training the SHL-DNN jointly with other highly-resourced language(s), such as English.

3. Indonesian Language

Indonesian language (Bahasa Indonesia) is a language officially used in Republic of Indonesia and spoken by almost 198 million people (2010) either as a first or as a second language [24]. It belongs to Austronesian group and is a variant of Malay language [25]. The language is used as a lingua franca in the archipelagic republic the people of which comprise hundreds of ethnicities and speak hundreds regional languages. Unlike Mandarin Chinese, it is not a tonal language. It is an SVO (Subject-Verb-Object) language. While the formal language usually follows the rule rigidly, the informal one does not. In the informal language, the important rule is the object always follows the verb. The change of the word order usually indicates the change of focus or emphasis. It is written in the 26-letter Latin alphabet. Each letter derives its name from its Dutch counterpart since Indonesia used to be a Dutch colony. The complete list of the name of each letter can be viewed in [26]. In Indonesian, almost every grapheme corresponds to one phoneme [27]. Table 3 lists all phonemes that occur in Indonesian and gives the mapping between each phoneme and its own grapheme based on [25]. The phoneme symbols used are the IPA (International Phonetic Alphabet).

When plosive /k/ occurs at the end of a native syllable, it will be realized as either /k/ or /ʔ/ depending on the speaker's ethnicity. It does not happen when the word is a loan word, e.g. the ‹k› in *fakta* is always pronounced /k/. Phonemes /f/, /z/, /x/, and /ʃ/ only occur in loan words especially from Arabic. Phoneme /v/ is almost always realized as either /f/ or /p/, hence there is no /v/ in the table. The glottal stop /ʔ/ can occur in four cases [25]. First, it is an allophone of /k/ as explained before. Second, it occurs between two vowels in some Arabic loan words such as *maaf* /maʔaf/. Third, it occurs between a vowel-final prefix and a vowel-initial root word such as *keamanan* /kəʔamanan/. Fourth, it occurs between a root word that ends in an /a/ and a suffix that starts with an /a/ such as *pertigaan* /pərtigaʔan/. The diphthongs only occur in an open position in a syllable, i.e. when the syllable does not end in a consonant phoneme.

Many Indonesian words can be formed with affixation, hence it belongs to agglutinative language family [28]. For example, the word *hasil* (result) can form words *berhasil* 'succeed', *keberhasilan* 'success', *ketidakberhasilan* 'failure', *menghasilkan* 'produce', *penghasil* 'producer', and *penghasilan* 'income'. Because of the affixation, a long word can be formed such as *kumempertanggungjawabkannya* 'I am responsible for it' although it is rare. There is no declension, conjugation, and tense for all words. Plural nouns are expressed by word reduplication. Indonesian also heavily borrows words form other languages, especially English and Arabic. Many English words are borrowed because of science and technology, while many Arabic loan words are because the majority of Indonesian people practice Islam.

4. The Training and Evaluation Data

4.1. Speech training and evaluation data description

There are three speech corpora used in this research. The first is an Indonesian dictated-speech corpus from [5] which was also reused in [14]. The second is the English *WSJ SI284* corpus. The third is an Indonesian spontaneous-speech corpus from [29]. Every corpus was

recorded in an acoustically clean environment into files that are monaural, formatted in WAV, and encoded using *Microsoft PCM*. They also have sampling rate of 16 kHz and bit rate of 256 kbps.

The Indonesian dictated-speech corpus comprises recordings from 20 native Indonesians, 11 are male and 9 are female. Every person reads not more than 343 prepared sentences. Every person's recording is then segmented into sentences and there are some people that have fewer than 343 due to some deleted recording segments (sentences) because they either get repeated or contain too much noise.

The corpus is then divided into training, development, and evaluation set. The training set will be used in training the baseline GMM-HMM and the SHL-DNN system and is named *ID-Train*. It comprises 10 people (6 males and 4 females) chosen randomly and speak not more than the first 270 of the 343 sentences for a total of approximately 5.5 hours. The development set, named *ID-Dev*, will be used for early-stopping the DNN training. It comprises the same 10 people speaking not more than the next 30 sentences of the 343 sentences. The training set together with the development set (*ID-Train-Dev*) will also be employed to pre-train a DBN before fine-tuning it through DNN training. The evaluation set (*ID-Read*) will be used as the standard evaluation for each dictated evaluation scenario in the research. It comprises the other 10 people (5 males and 5 females) that speak not more than the last 43 of the 343 sentences for a total of 4,926 words. More details about the separation are given in Table 1 and Table 2.

The *WSJ SI284* corpus comprises 283 people (142 males and 141 females) speaking a total of 37,416 sentences. The sentences were taken from articles of the English Wall Street Journal. This corpus will be used exclusively as a pre-training and training set for the SHL-DNN jointly with the *ID-Train*.

The Indonesian spontaneous-speech corpus comprises 299 people, 146 are male and 153 are female. They speak a total of 18,806 sentences (270,478 words). Unlike the Indonesian dictated-speech data, its sentences were not prepared. In a clean recording environment, each person was asked about different topics and his/her response was recorded. The response was recorded per person and then manually segmented into sentences. Because there was no prepared transcript, each segment had to be manually annotated. Due to the nature of spontaneous speech, every segment could contain noises and fillers, such as breathing and hissing sound. The noises and fillers were annotated with tags in the transcript. This corpus will be used exclusively as the standard evaluation set for each spontaneous evaluation scenario in this research and is named *ID-Spontan*. Statistic of each corpus is summarized in Table 1 and Table 2.

Table 1. Detailed Information about Speech Training and Development Sets

Corpus	#Speakers	#Sentences
ID-Train	10	31,548
ID-Dev		3,340
WSJ SI284	283	37,416

Table 2. Detailed Information about Speech Evaluation Sets

Corpus	#Speakers	#Sentences	#Words
ID-Read	10	425	4,926
ID-Spontan	299	18,806	270,478

4.2. The phoneme set and the dictionary

The phoneme set and their representation are listed in Table 3. As seen in the table, the symbols are fewer than their own phoneme counterpart. The glottal stop /ʔ/ that is the allophone of /k/ is represented by symbol "k", while in other cases, the glottal stop is not represented by any symbol. Each diphthong is translated into its own vowel and approximant constituent based on its representation in [25]. For example diphthong /aj/ can be seen as phoneme /a/ + /j/ by its representation, hence it is represented as a double-symbol "a y" (notice the space between the two symbols). Both phoneme /e/ and /ə/ are represented with only one symbol "e". This is because both are written with one grapheme ‹e› and there is no clear rule on whether the grapheme ‹e› is pronounced as an /e/ or /ə/. Indonesians are sometimes confused about the rule themselves. For example, the word *macet* 'jammed' should be pronounced as /macət/ but many Indonesians

pronounce it as /macet/. Some ethnicities, such as Batak people, are also having difficulty in pronouncing the /ə/. Therefore, both phonemes are represented with only one symbol "e".

As seen in Table 3, there is a high degree of regularity between the graphemes and their own symbol(s). Based on the fact, a tool is built to help developing a speech dictionary by extracting all words that appear in the Indonesian dictated-speech corpus (the 343 prepared sentences) and translating all graphemes to their own correspondent phoneme(s) (grapheme-to-phoneme tool). The rules are as follows:

a. Each diphthong digraph is translated based on its own constituent letter to simplify the rule. This rule is introduced because each diphthong's appearance in a word cannot be predicted. This is not the case with other consonant digraphs. For example, the digraph ‹ng› is always pronounced as /ŋ/, hence it is always represented as "ng" (notice the lack of space). Therefore, for example, when grapheme ‹a› occurs, the next grapheme does not have to be investigated, but when ‹n› occurs, the next grapheme must be investigated to determine whether the translation is "n", "ng", or "ny".
b. The glottal stop, except for the allophone of /k/, is not translated to any symbol.
c. Except for the diphthongs and the glottal stop, all graphemes are translated to their own symbol based on Table 3.

Table 3. List of Mapping between Indonesian Phonemes, Graphemes, and their own HMM Training Symbol

Phonetic Category	Phoneme	Usual Grapheme	Symbol	Example
Vowel	/a/	‹a›	a	**a**da
	/e/	‹e›	e	**e**nak
	/ə/			**e**mas
	/i/	‹i›	i	**i**si
	/o/	‹o›	o	**o**bat
	/u/	‹u›	u	**u**rus
Diphthong	/aj/	‹ai›	a y	ab**ai**
	/aw/	‹au›	a w	engk**au**
	/oj/	‹oi›	o y	amb**oi**
Plosive	/b/	‹b›	b	**b**ibi
	/d/	‹d›	d	**d**ada
	/g/	‹g›	g	**g**agap
	/k/	‹k›	k	**k**akak
		‹q›		**Q**uran
	/p/	‹p›	p	**p**aku
	/t/	‹t›	t	**t**ata
	/ʔ/	‹›		maaf *)
		‹k›	k	ta**k**
Affricate	/tʃ/	‹c›	c	**c**ucu
	/dʒ/	‹j›	j	**j**aja
Nasal	/m/	‹m›	m	**m**asa
	/n/	‹n›	n	**n**ama
	/ɲ/	‹ny›	ny	**ny**anyi
	/ŋ/	‹ng›	ng	**ng**eri
Trill	/r/	‹r›	r	**r**asa
Fricative	/f/	‹f›	f	**f**asih
		‹v›		**v**ia
	/h/	‹h›	h	**h**ari
	/x/	‹kh›	kh	a**kh**ir
	/s/	‹s›	s	**s**aya
	/ʃ/	‹sy›	sy	**sy**arat
	/z/	‹z›	z	**z**akat
Approximant	/w/	‹w›	w	**w**aktu
	/j/	‹y›	y	**y**akin
Lateral Approximant	/l/	‹l›	l	**l**aut

Note: *) the phoneme occurs between the two a's in *maaf*

d. One rule that is not included in Table 3 is introduced, i.e. grapheme ‹x› is translated to "k s".

Because the rule is simple, many discrepancies between the resulting phoneme sequence and the actual pronunciation are introduced to the end result. Therefore, every entry in the resulting dictionary is then manually checked against its actual pronunciation. There are

some cases where correction needs to be performed to the phoneme sequence, which are as follows:

a. The diphthong case explained before; a grapheme sequence that should be pronounced as a diphthong is translated to its own symbol, e.g. ‹ai› that is pronounced as the diphthong /aj/ is translated to "a y".
b. The resulting phoneme sequence for an abbreviation is corrected to pronunciation of its own individual constituent letters, e.g. ‹abg› is translated to "a b e g e" not "a b g". Pronunciation of each letter is based on [26].
c. Since grapheme ‹x› comes from foreign words, its pronunciation is not uniform. Not all ‹x› are pronounced as "k s"; there are also some ‹x›-es that are pronounced as "s" since it is impossible to pronounce the /k/. For example, it is impossible to pronounce the /k/ in *xanana* (a foreign name), so it is pronounced as "s a n a n a" rather than "k s a n a n a".
d. The grapheme ‹y› behaves somewhat ambiguously in Indonesian. It can be pronounced as a /j/ especially in native words, e.g. in *ayah* /a y a h/ 'father'. However, it is sometimes pronounced as an /i/ especially in foreign names and words, e.g. in *tommy* /t o m m i/. Therefore, it must be investigated on per word basis.
e. The grapheme ‹h› is sometimes silent. Again, this comes from foreign names and words like *manchester* "m a n c e s t e r".
f. The digraph ‹sh› is sometimes pronounced as a "sy" but sometimes as two separate phonemes "s h". Again, the former occurs mainly in foreign names and words.
g. Other cases where it is impossible to pronounce the phoneme(s). For example *solowiejczyk* is translated to "s o l o w i e z i k" as it is impossible to pronounce the /dʒ/ (‹j›) and /tʃ/ (‹c›) phoneme.

Although in the *ID-Train*, *ID-Dev*, and *ID-Test*, foreign names and words are pronounced according to the Indonesian rule of pronunciation (not their original language), there are still some words that cannot be pronounced the way Indonesian pronunciation rule stipulates. The corrections are made to accommodate such foreign names and words. The resulting dictionary has 2,321 words after the addition of "‹unk›" entry for unknown words. The entries are all lowercased to simplify the training and decoding process. Symbol SIL is also appended to the list of phonemes; it signifies all silences, noises, and unknown words.

4.3. The text corpus and the language model

The text corpus for building the language model (LM) was also obtained from [5]. It was compiled from *Kompas* newspaper and *Tempo* magazine online collections. In this research, it is named "Tala" after its compiler. After some cleanings (the details can be viewed in [5]), the resulting text corpus contains 613,054 sentences. The detailed statistic for the text corpus is given in Table 4.

A 3-gram LM is built using the SRILM Language Modeling Toolkit [30] augmented with Kneser-Ney smoothing and interpolation of higher-order with lower-order probability estimates [31]. Vocabulary for building the LM comprises all words that appear in the dictionary. The LM is then evaluated using the *ID-Read* and the *ID-Spontan* transcriptions. For the *ID-Read*, only the last 43 sentences are used, not all sentences. The resulting perplexities are shown in Table 5.

Table 4. Detailed Statistics about the "Tala" Text Corpus

Attribute	Quantity
Number of sentences	613,054
Number of words	10,250,637
Average words per sentences	16.72
Number of unique words	108,224

Table 5. Perplexities of Each Text Corpus Evaluation Set

Evaluation Set	Perplexity	#OOV/#Unique Words
ID-Read (43 sent.)	161.74	0/499 (0%)
ID-Spontan	231.84	116,537/269,631 (43.22%)

Note: OOV is out-of-vocabulary words, i.e. words that do not appear in the vocabulary (dictionary)

5. Experiments and Results

5.1. Feature extraction

To train an HMM-DNN model, a GMM-HMM model must first be trained. For training the GMM-HMM model, 13-dimension MFCC features (the 0^{th}-12^{th} cepstral coefficient) are extracted from each of the speech training and evaluation sets. The Δ and ΔΔ features are appended only in the monophone and triphone training.

To train the DNN, 40-dimension mel filterbank features are extracted from each of the training, development, and evaluation sets. As mentioned in Section II.A, a DNN can be trained with highly-correlated features more efficiently than a GMM-HMM and the resulting recognizer yields fewer errors than when it is trained using the MFCCs. To obtain MFCC features, mel filterbank features must be extracted first then be decorrelated using the discrete cosine transform (DCT) [32]. Therefore, mel filterbanks are less decorrelated than MFCCs.

Both feature extraction methods are followed by per-speaker cepstral mean and variance normalization (CMVN). CMVN is claimed to make the extracted features more robust to environmental noises by normalizing the means and variances of each speaker's features to zero [33]. All experiment steps including the feature extractions and the model trainings will be conducted using the Kaldi toolkit for speech recognition [34].

5.2. Building the GMM-HMM models

A DNN requires target values. In a DNN for recognizing speech, its target values are posterior probabilities generated by the Viterbi alignment using a GMM-HMM model. Therefore, a GMM-HMM is required to train the DNN. In this research, two GMM-HMM models are required, one for Indonesian and one for English. Both Indonesian and English GMM-HMM will be required for the forced alignment to produce posterior probabilities used as the training target for the SHL-DNN later. In addition, the Indonesian one will also be required to produce baseline evaluation results to be compared to the results of the SHL-DNN. The Indonesian GMM-HMM models will be trained with the *ID-Train* set while the English models with the *WSJ SI284* corpus. Steps to build the GMM-HMM models as the training target for the DNN are essentially similar to Type-I features experiment of [35] with some minor changes.

The first step is to train two monophone models (one for each language) using the obtained MFCC features appended with their own first and second derivatives (MFCC + Δ + ΔΔ). Each model is trained with its respective training data. Each resulting model is then employed in forced-alignment process required for its respective triphone training. Next, two triphone models are trained with the appended MFCC features. Each model is specified to have 2000 leaves and 10000 mixtures; Kaldi can only specify total number of mixtures and cannot specify number of mixtures each leaf has.

Using force-alignment results by the triphone models, two LDA/MLLT (linear discriminant analysis followed by maximum likelihood linear transform) models are trained on only the 13-dimension MFCCs (not using the Δ and ΔΔ). Each model is to have 2500 leaves and 15000 mixtures. For training each LDA model, each 13-dimension MFCC frame is spliced across ± 3 frames resulting in a 91-dimension vector. The LDA [36] is then applied to all spliced vectors, decorrelating them and reducing their dimensions to 40. The GMM-HMM states are used as the classes for the LDA estimations. The resulting 40-dimension vectors are then used to estimate the MLLT [37] (also known as "global semi-tied covariance" transformation [38]) for the GMMs. The MLLT transforms the GMMs to model the feature distributions more accurately. As described, the LDA/MLLT improve the modeling process from two sides; the LDA improves the features and the MLLT improves the models.

Same as before, the LDA/MLLT models are employed in the forced alignment for training the next set of models, i.e. two SAT (speaker adaptive training) models. As with the LDA/MLLT models, each SAT model is also to have 2500 leaves and 15000 mixtures. SAT is an estimation process where two maximum-likelihood linear regressions (MLLRs) are employed, one to the feature space in form of fMLLR (feature-space MLLR) and one to the model space in form of CMLLR (constrained MLLR) [39]. The GMMs are transformed by CMLLR using training data that have been transformed by fMLLR. The Indonesian SAT model will then be employed to decode each evaluation set, the *ID-Read* and the *ID-Spontan*. The results will be used as a baseline for decoding results of the SHL-DNN. Forced alignment using SAT model for each language will then be employed as the training target for the SHL-DNN. The final Indonesian

SAT model produces 1,933 senones (context-dependent HMM states), while the English model produces 3,367 senones.

5.3. Pre-training the DBN and training the SHL-DNN

A fully-connected DBN is first pre-trained to become the shared layers in the SHL-DNN. As mentioned before, it is not necessary for the DBN to know which language the training data belong to. Therefore the DBN can be built efficiently using training data for both languages without changing anything. To pre-train it, the Indonesian training data *ID-Train* are first merged with the development data *ID-Dev* to become *ID-Train-Dev*. The *ID-Train-Dev* are then merged with the English *WSJ SI284* set and shuffled. The shuffling is performed to distribute the Indonesian training data uniformly inside the English data, so the pre-training will not skew to either language. The resulting set will be used as pre-training data for the DBN.

The DBN will have 6 hidden layers, each has 2048 hidden units. Each hidden unit has logistic activation function the output of which ranges from 0 to 1. The input to the DBN is a 40-dimension mel filterbank frame spliced across ± 5 frames (11 adjacent frames). The trained DBN is then appended with a block softmax output layer to form the SHL-DNN. The block softmax layer contains two layers of softmax units, one layer for Indonesian and one for English. The two layers share the same hidden layers, but connections from one softmax layer to the hidden layers are independent from another softmax layer. The output layer represents posterior probability of each corresponding HMM state. The Indonesian softmax layer contains 1,933 units corresponding with number of senones outputted by the Indonesian triphone + LDA/MLLT + SAT model, while the English layer contains 3,367 units. Architecture of the SHL-DNN in this experiment is based on Figure 2 with Indonesian as “Language 1” and English as “Language 2”.

The training is performed with only the shuffled Indonesian *ID-Train* set merged with the English *WSJ SI284* set without the *ID-Dev* set. The *ID-Dev* are utilized for early-stopping the DNN training to prevent overfitting. Input to the SHL-DNN is also a 40-dimension mel filterbank frame spliced across ± 5 frames (11 adjacent frames). The whole SHL-DNN is trained using cross-entropy (CE) as its cost function and an initial learning rate of 0.008. After some training epochs, the learning rate will be halved for each subsequent epoch. The training will stop after the cost reduction is less than 0.1% of the previous epoch or maximum number of training epochs is reached.

The training target is the forced alignment generated by the triphone + LDA/MLLT + SAT model for each language. This is the reason the triphone + LDA/MLLT + SAT model is also required for English. Actual outputs of softmax layer for one language will be compared to the corresponding forced-alignments for the language to determine the cost. Discrepancies between the actual outputs and the forced alignments of one language will update the hidden layers of the SHL-DNN and only the sofmax layer for the corresponding language. After the training is finished, the English softmax layer is discarded because it is not required for decoding, leaving only the 1,933-unit Indonesian output layer. The SHL-DNN will decode each Indonesian evaluation set (*ID-Read* and *ID-Spontan*). The result for each set will be compared to the decoding result of the corresponding set using the Indonesian triphone + LDA/MLLT + SAT model.

5.4. Results

Results are reported in terms of word error rate (WER). The WER is defined as

$$WER = \left(\frac{S + I + D}{N} \right) \times 100\%. \quad (5)$$

where S is number of substituted words in results compared to their reference, I is number of inserted words, D is number of deleted words, and N is total number of words in the reference sentences. The lower the WER the better. The WER for each evaluation set decoded by the GMM-HMM triphone+LDA/MLLT+SAT model and the SHL-DNN is given in Table 6. Based on the table, it can be seen that the SHL-DNN always outperforms the GMM-HMM model, even in the *ID-Spontan* decoding where the number of OOV words are high.

Table 6. Decoding Results in Terms of WER (%)

Evaluation Set	GMM-HMM	SHL-DNN
ID-Read	7.94	4.12
ID-Spontan	70.62	66.43

6. Analysis and Discussion

The SHL-DNN outperforms the GMM-HMM on the decoding result for both the dictated and spontaneous evaluation set. This confirms the effectiveness of jointly training a DNN for an under-resourced target language with another highly-resourced language although the language is distantly related with the target language. In the decoding result of the dictated evaluation set *ID-Read*, it is understandable that the WER is low because the dictionary is built using only the words that appear in the *ID-Train*, *ID-Dev*, and *ID-Read*. This makes the OOV nil and the word searching space not too big. However, what makes the SHL-DNN outperform the GMM-HMM is that the SHL-DNN can successfully decode some named entities. Those include foreign-sounding names (e.g. *yevgeny*, *islamabad*), native-sounding names (e.g. *joko*, *tenau*), and native names with old spelling (e.g. *suadmodjo*). The GMM-HMM also fails to decode some abbreviations, like *ntt* (abbreviation of East Nusa Tenggara province in Indonesia) and *uu* (abbreviation of *Undang-Undang* 'act/law'). The GMM-HMM model decodes those words to similar word(s), e.g. *yevgeny* /yefgeni/ is decoded as "*di f g ni*" /di ef ge ni/. Although there are also some words that are unsuccessfully decoded by both models (e.g. *jankulovski*), those cases are more pronounced in the decoding result by the GMM-HMM than the SHL-DNN. That makes insertion errors and deletion errors higher in the GMM-HMM result than in the SHL-DNN one.

The decoding on spontaneous evaluation set *ID-Spontan* yields high WER because the OOV rate is high (43.22%). This makes OOV words either be decoded as other word(s) or be deleted. Because the *ID-Spontan* are more recent than the *ID-Read*, the OOV words also include words that recently are more frequently spoken, such as *jokowi*, name of the recently elected Indonesian president. In the decoding results by both models, fillers and noises are decoded almost unsuccessfully for all sentences. This is because there is no example of filler and noise in both English and Indonesian (dictated) training data. Both training sets were recorded in an acoustically clean environment. Looking at the resulting sentences, the GMM-HMM decoding result has more insertion errors than the SHL-DNN result. That is the reason the GMM-HMM yields higher WER than the SHL-DNN since number of insertion errors has no upper limit. In both decoding scenarios, insertion errors are also directly affected by Indonesian grammar. Since Indonesian grammar frequently uses reduplication, the reduplication is decoded to two indentical words increasing insertion errors. Some words are also decoded to other similar sounding word(s) increasing the substitution and insertion errors. For example, *pengangguran* is decoded to *penangguhan* (similar sounding word increasing substitution errors) and *inflasi* is decoded to "*indra si*" (similar sounding two words increasing substitution and insertion errors).

7. Conclusion

The results show that augmenting training data for an under-resourced language, such as Indonesian, with another highly-resourced language, such as English, and using them to train an SHL-DNN decreases the WER. Shuffled training data for both languages can be utilized as pre-training data for a DBN efficiently without any difference with a monolingual DBN pre-training. The resulting DBN can be appended with a softmax layer for each language and be trained simultaneously with training data for both languages in form of SHL-DNN. The SHL-DNN is then employed to decode Indonesian dictated and spontaneous speech. The SHL-DNN decoding on the dictated speech yields 3.82% absolute (48.11% relative) decrease in WER compared to the monolingual Indonesian GMM-HMM while the decoding on the spontaneous speech yields 4.19% absolute (5.93% relative) decrease. This shows the effectiveness of a multilingual SHL-DNN for the under-resourced Indonesian language compared to a monolingual GMM-HMM.

Acknowledgment

This research was partially supported by the Master's Program toward Doctoral Degree for Excellent Graduate (Program Pendidikan Magister Menuju Doktor untuk Sarjana Unggul/PMDSU) from Kemenristekdikti Indonesia within research entitled "Indonesian Automatic Speech Recognition System". We also would like to thank PT INTI, Bandung, Indonesia for the utilization of the spontaneous speech corpus presented in this research.

References

[1] Besacier L, Barnard E, Karpov A, Schultz T. Automatic speech recognition for under-resourced languages: A survey. *Speech Commun*. 2014; 56: 85-100.
[2] Gales MJ, Knill KM, Ragni A, Rath SP. *Speech recognition and keyword spotting for low resource languages: BABEL project research at CUED*. SLTU. St Petersburg. 2014: 16-23.
[3] Sahraeian R, Compernolle DV, Wet Fd. *Under-resourced Speech Recognition based on the Speech Manifold*. INTERSPEECH. Dresden. 2015: 1255-1259.
[4] Hoesen D, Satriawan CH, Lestari DP, Khodra ML. *Towards robust Indonesian speech recognition with spontaneous-speech adapted acoustic models*. SLTU. Yogyakarta. 2014: 167-173.
[5] Lestari DP, Iwano K, Furui S. *A large vocabulary continuous speech recognition system for Indonesian language*. 15th Indonesian Scientific Conference in Japan. Tokyo. 2006.
[6] Imseng D, Motlicek P, Bourlard H, Garner PN. Using out-of-language data to improve an under-resourced speech recognizer. *Speech Commun*. 2014 Jan; 56: 142-151.
[7] Burget L et al. *Multilingual acoustic modeling for speech recognition based on subspace Gaussian mixture models*. ICASSP. Dallas. 2010: 4334-4337.
[8] Heeringa W, Wet Fd. *The origin of Afrikaans pronunciation: a comparison to West Germanic languages and Dutch dialects*. 19th Annual Symposium of the Pattern Recognition Association of South Africa. Cape Town. 2008: 159-164.
[9] Beekes RSP, Vaan Md. Comparative Indo-European linguistics: An introduction. 2nd ed. Amsterdam: John Benjamins; 2011.
[10] Martin T, Svendsen T, Sridharan S. *Cross-lingual pronunciation modelling for Indonesian speech recognition*. EUROSPEECH. Geneva. 2003: 3125-3128.
[11] Hinton G et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Process. Mag.* 2012; 29(6): 82–97.
[12] Ciresan DC, Meier U, Gambardella LM, Schmidhuber J. Deep, big, simple neural nets for handwritten digit recognition. *Neural Comput*. 2010 Dec; 22(12): 3207–3220.
[13] Huang JT, Li J, Yu D, Deng L, Gong Y. *Cross language knowledge transfer using multilingual deep neural network with shared hidden layers*. ICASSP. Vancouver. 2013: 7304–7308.
[14] Hoesen D, Price R, Lestari DP, Shinoda K. *A DNN-based ASR system for the Indonesian language.* ASJ Autumn Meeting. Aizu-Wakamatsu. 2015: 5-6.
[15] Nakamura M, Iwano K, Furui S. Differences between acoustic characteristics of spontaneous and read speech and their effects on speech recognition performance. *Comput. Speech Lang*. 2008; 22(2): 171-184.
[16] Ward W. *Understanding spontaneous speech*. Speech and Natural Language: Proceedings of a Workshop. Philadelphia (PA). 1989: 137-141.
[17] Rumelhart DE, Hinton GE, Williams RJ. *Learning representations by back-propagating errors.* Nature. 1986; 323(6088): 533–536.
[18] Glorot X, Bengio Y. *Understanding the difficulty of training deep feed forward neural networks.* AISTATS. Sardinia. 2010: 249–256.
[19] Dahl GE, Yu D, Deng L, Acero A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans. Audio Speech Lang. Process.* 2012; 20(1): 30-42.
[20] Hinton GE, Osindero S, Teh Y. A fast learning algorithm for deep belief nets. *Neural Comput.* 2006; 18(7): 1527–1554.
[21] Hinton GE, Salakhutdinov R. Reducing the dimensionality of data with neural networks. *Science*. 2006; 313(5786): 504–507.
[22] Young S et al. The HTK Book (for HTK Version 3.4). Cambridge: Cambridge University Engineering Department; 2009.
[23] Mohamed A, Dahl G, Hinton GE, Acoustic modeling using deep belief networks. *IEEE Trans. Audio Speech Lang. Process*. 2012 Jan; 20(1): 14–22.
[24] BPS - Statistics Indonesia. Population of Indonesia: Result of Population Census 2010. Jakarta: BPS - Statistics Indonesia; 2012.
[25] Soderberg CD, Olson KS. *Indonesian. J. Int. Phon. Assoc*. 2008; 38(2): 209-213.
[26] Panitia Pengembang Pedoman Bahasa Indonesia, Kementerian Pendidikan dan Kebudayaan. Pedoman Umum Ejaan Bahasa Indonesia. 4th ed. Jakarta: Badan Pengembangan dan Pembinaan Bahasa; 2016.

[27] Yap MJ, Liow SJ. The Malay lexicon project: A database of lexical statistics for 9,592 words. *Behav. Res. Methods*. 2010; 42(4): 992-1003.
[28] Sakti S, Hutagaol P, Arman AA, Nakamura S. *Indonesian speech recognition for hearing and speaking impaired people*. ICSLP. Jeju, South Korea. 2004: 1037–1040.
[29] Hoesen D, Lestari DP, Khodra ML. *Adaptation of acoustic model for Indonesian using varying ratios of spontaneous speech data*. OCOCOSDA. Denpasar. 2016: 39–44.
[30] Stolcke A. *SRILM – An extensible language modeling toolkit.* ICSLP. Denver (CO). 2002: 901–904.
[31] Chen SF, Goodman J. An empirical study of smoothing techniques for language modeling. *Comput. Speech Lang.* 1999; 13(4): 359-394.
[32] Logan B. *Mel frequency cepstral coefficients for music modeling*. ISMIR. Plymouth (MA). 2000.
[33] Viikki O, Laurila K. Cepstral domain segmental feature vector normalization for noise robust speech recognition. *Speech Commun*. 1998; 25(1-3): 133-147.
[34] Povey D et al. *The Kaldi speech recognition toolkit*. ASRU. Hawaii. 2011.
[35] Rath SP, Povey D, Veselý K, Černocký J. *Improved feature processing for Deep Neural Networks*. INTERSPEECH. Florence, Italy. 2013: 109-113.
[36] Duda RO, Hart PE, Stork DG. Pattern classification. 2nd ed. New York: Wiley; 2000.
[37] Gopinath RA. *Maximum likelihood modeling with Gaussian distribution for classification*. ICASSP. Seattle (WA). 1998; 2: 661-664.
[38] Gales MJ. Semi-tied covariance matrices for hidden Markov models. *IEEE Trans. Audio Speech Lang. Process*. 1999; 7(3): 272–281.
[39] Gales MJ. Maximum likelihood linear transformations for HMM-based speech recognition. *Comput. Speech Lang.* 1998; 12(2): 75–98.