Radzikowski *et al. EURASIP Journal on Audio, Speech, and Music Processing* (2019) 2019:3
https://doi.org/10.1186/s13636-018-0146-4

EURASIP Journal on Audio, Speech, and Music Processing

RESEARCH Open Access

Dual supervised learning for non-native speech recognition

Kacper Radzikowski[1*], Robert Nowak[1], Le Wang[2] and Osamu Yoshie[2]

Abstract

Current automatic speech recognition (ASR) systems achieve over 90–95% accuracy, depending on the methodology applied and datasets used. However, the level of accuracy decreases significantly when the same ASR system is used by a non-native speaker of the language to be recognized. At the same time, the volume of labeled datasets of non-native speech samples is extremely limited both in size and in the number of existing languages. This problem makes it difficult to train or build sufficiently accurate ASR systems targeted at non-native speakers, which, consequently, calls for a different approach that would make use of vast amounts of large unlabeled datasets. In this paper, we address this issue by employing dual supervised learning (DSL) and reinforcement learning with policy gradient methodology. We tested DSL in a warm-start approach, with two models trained beforehand, and in a semi warm-start approach with only one of the two models pre-trained. The experiments were conducted on English language pronounced by Japanese and Polish speakers. The results of our experiments show that creating ASR systems with DSL can achieve an accuracy comparable to traditional methods, while simultaneously making use of unlabeled data, which obviously is much cheaper to obtain and comes in larger sizes.

Keywords: Speech recognition, Dual supervised learning, Reinforcement learning, Policy gradients, Non-native speaker, Machine learning, Deep learning, Artificial intelligence

1 Introduction

Speech recognition has been the subject of extensive research since the second half of the previous century. Its main purpose is to allow communication between a human and a machine, using the most natural way for a human to convey a message—speech.

The speech recognition techniques and methodologies that have been developed recently can work with up to 90–95% accuracy, depending on the dataset and benchmark test used [1]. However, such accuracy levels can be reached only when the system is used for recognizing the speech of native speakers (e.g., English language for North American people). In the case of non-native speakers, even the most advanced speech recognition systems can only achieve an accuracy of up to 50–60%. The main reason for such a drop is that non-native speakers have a different mother tongue than the one that is being recognized. Usually, the language used most often by a person is his or her mother tongue, and the pronunciation of this language, with its patterns and characteristics, affect the pronunciation of a foreign language, causing the failure of speech recognition systems. However, global integration creates the need to properly recognize non-native speakers, who nowadays represent the vast majority of users.

2 Methods

2.1 Problems with traditional methodology

The easiest way for speech recognition systems to achieve higher accuracy with non-native speakers would be to train a classifier for speech recognition for a specific language and nationality/ethnic group of non-native speakers of that language [2, 3].

However, this idea is not feasible in most real world cases. The reason for this is the size of available speech datasets. In traditional methods of training speech recognition classifiers, supervised learning techniques are usually applied. Those require labeled datasets of a large size. While perfectly fitted for recognizing the speech of

*Correspondence: radzikowskikacper@gmail.com
[1]Warsaw University of Technology, Institute of Computer Science, Warsaw, Poland
Full list of author information is available at the end of the article

tens of the most popular languages worldwide, supervised learning techniques do not provide classifiers of a decent quality for non-native speech. The main reason for this problem concerns the size of speech datasets for a certain language. Even if they exist, the number of samples is usually not large enough to build an acoustic model which could reflect the real-world distribution of speech signal characteristics in one particular language. Additionally, the vocabulary in such databases comprises usually not more than a few thousand words, while a typical dictionary contains at least tens of thousands of words. Moreover, attempting to train a speech recognition classifier for one language and one nationality/ethnic group of non-native speakers would require a new database, which would involve a large workforce and budget. For these reasons, traditional methods of training classifiers for the purpose of speech recognition are usually not applicable for non-native speech [4–11].

In comparison to labeled datasets, unlabeled datasets are both much more easily available and larger in size for many ethnic groups speaking a second language. This vast amount of unlabeled data could theoretically be used to develop a method for training classifiers in the recognition of non-native speech.

Our research hypothesis states that it is possible to create a method that uses unlabeled datasets of two speech-related domains: speech samples without corresponding transcripts and text corpora without corresponding speech samples, to train speech recognition classifiers in a way which is as efficient and accurate as training methods provided by traditional solutions. The unlabeled data used in our method is far cheaper and easier to obtain and it usually comes in larger amounts than labeled data required by the traditional methods that have been widely used until now. The methodology we used in our experiments is based on the dual supervised learning (DSL) technique [12]. It exploits the fact that speech recognition and speech synthesis are complementary to each other.

2.2 Methodology used in this research

DSL is a concept introduced by Xia [12]. It is based on the acknowledgement that numerous supervised learning tasks emerge in dual forms (e.g., English-to-French and French-to-English translation, speech recognition and synthesis, image classification and image generation, etc.). The dual tasks have intrinsic connections to each other due to the probabilistic correlation between their models.

To exploit the duality, a new learning scheme which involves two tasks—a primal task and its dual task—can be formulated. The primal task takes a sample from space X as input and maps to space Y, and the dual task takes a sample from space Y as input and maps to space X. Using the language of probability, the primal task learns a conditional distribution $P(y|x;\theta_{xy})$ parameterized by θ_{xy}, and the dual task learns a conditional distribution $P(x|y;\theta_{yx})$ parameterized by θ_{yx}, where $x \in X$ and $y \in Y$. In the new scheme, the dual tasks are jointly learned and their structural relationship is exploited to improve the learning effectiveness.

DSL for machine translation (e.g., English to French) has already been tackled successfully [12]. The researchers have shown that it is possible to create a similar algorithm which could train a fully functional and accurate translation system using the dual characteristics of the problem. In our study, we employed and adjusted this methodology to the domain of text and sound: text to speech (TTS) and speech to text (STT).

The idea is based on reinforcement learning algorithms, which do not require data in the same form that supervised learning does. All we need are two unlabeled datasets. One dataset is a set of speech recordings by non-native speakers of L language who belong to N nationality. The second one is text corpora of the L language.

2.3 Applied models

We are going to exploit the easy access to unlabeled datasets in order to train two separate models. One model is a language one (M_L). It is created solely with a text corpus. There are two required functionalities: (1) the possibility to generate a new sentence in textual form in that language and (2) the possibility to estimate a probability score for a given sentence in that language (i.e., how natural a given sentence is, according to the language model).

The second model is an acoustic one (M_S). It is created with only unlabeled speech recording datasets. We would like it to have a similar functionality as the first model, but for the speech domain. Namely, we want it to be able to synthesize a new recording from the represented sound distribution as well as estimate the probability score for a given sound sequence, saying how accurately the sound sequence can be recognized as speech according to the acoustic model. The two models were trained separately, in isolation from any other models, during the separate tasks of language modeling and acoustic modeling, respectively. The DSL methodology was not yet used at this point. In the training processes of the two models, only unlabeled datasets were utilized. In the language model, a text corpus was used. In the acoustic model, a set of speech recordings was used. After training, each of those models had the ability to generate a random sample from the learned probability distribution and to estimate a likelihood score of a sample, with respect to the learned probability distribution. In the language model, the sample becomes a textual sentence, and in the acoustic model, the sample is a soundwave, a recording of a speech.

The setup of this method contains two more models. The first one is a speech recognition model (M_{STT}), which can recognize phonemes for a given sound sequence. The other one, complementary to the first, is a speech generation model (M_{TTS}), with the functionality of generating a speech signal for a given textual sentence. In the process of training using the DSL approach (described later), these two latter models (M_{STT} and M_{TTS}) will be the only trainable ones. They will be initialized by means of either a warm-start or a semi warm-start mode and will have their weights updated according to a gradient descent-based algorithm.

The two former models (M_L and M_S) were trained before starting the DSL-based training process. They were trained in isolation from any other models, using unlabeled datasets. Therefore, they did not take part in the DSL-based training process.

Our method uses all four aforementioned models, closed in a feedback loop.

The role of each model is crucial, because each of them is responsible for either synthesizing new data samples, evaluating the results yielded by the previous model, or converting the data between the textual and acoustic domain. Having stated that, we find that both the language and acoustic models have an ability to generate a new data sample in the form of a textual sentence or recording, respectively. Moreover, they can estimate the correctness (by giving a likelihood score) for a given sample using the learned probability distribution of data in their domain (either a text corpus or recording datasets). Due to that, these models can give feedback to the model which converts the data between domains, making it possible for the model to learn weights which will lead to better (in terms of the feedback-giving model) conversion results during the next iteration of the training process.

2.4 Feedback loop

In the process of training, we decided to make use of two kinds of loop.

The first type (called loop L) is depicted in Fig. 1. The loop begins from a language model M_L generating a t sentence in text form.

Then, speech generation model (M_{TTS}) generates sound samples which can potentially represent how the t sentence may sound when pronounced, according to M_{TTS}. M_{TTS} generates K different soundwaves $TTS(t)_k$ from t sentence, using a beam search algorithm.

The third step is a probability estimation for each of the K generated samples. This is achieved by utilizing the acoustic model M_S. The score for each sample equals:

$$a_k^{im} = M_S(TTS(t)_k) \tag{1}$$

where:

$$\begin{aligned} a_k^{im} &= \text{immediate reward score for } k \text{ sample} \\ &\quad \text{of soundwave} TTS(t) \text{ for loop} L \\ M_S(TTS(t)_k) &= \text{likelihood score for} k \text{ sample} \end{aligned}$$

This says how "probable" it is that the synthesized recording could be an actual speech sample in a particular language.

Lastly, the speech recognition model (M_{STT}) transfers a previously synthesized sample $TTS(t)_k$ into textual form. At this step, we also calculate a probability score for each of the K synthesized samples that says how correctly the M_{STT} model recognizes the k sample as the original sentence t. The score equals:

$$a_k^{lt} = logP(t|TTS(t)_k; M_{STT}) \tag{2}$$

where:

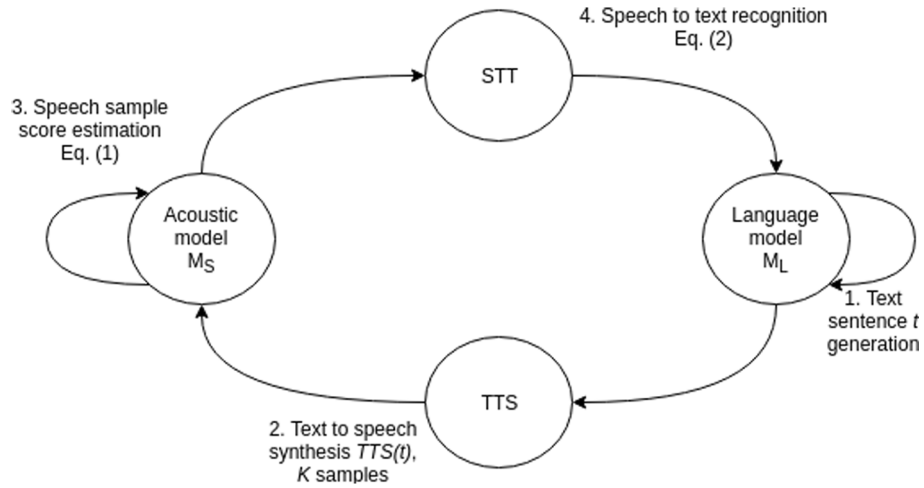

Fig. 1 Feedback loop design. Loop starts with generating a sentence (loop *L*)

$$
\begin{aligned}
a_k^{lt} &= \text{long-term reward score for} k \\
&\quad \text{speech sample of} TTS(t) \text{ for loop} L \\
P(t|TTS(t)_k; M_{STT}) &= \text{probability score for receiving} \\
&\quad \text{sentence} t \text{ from} k \text{ speech sample} \\
&\quad TTS(t), \text{ when recognizing using} M_{STT}
\end{aligned}
$$

The second kind of loop is similar to the first, but starts at another point. It is shown in Fig. 2 and is called loop *S*.

This loop begins from the acoustic model M_S generating a speech sample *s*.

Then, M_{STT} recognizes a generated sample as textual sentences which are potentially transcripts for *s* sample, according to M_{STT}. M_{STT} produces *K* most probable sentences $STT(s)_k$ from *s* sample, also using a beam search algorithm.

The third step is a probability score estimation for each of the *K* recognized sentences. This is achieved by applying the language model M_L. The score for each sentence equals:

$$b_k^{im} = M_L(STT(s)_k) \tag{3}$$

where:

$$
\begin{aligned}
b_k^{im} &= \text{immediate reward score for} k \\
&\quad \text{sentence of} STT(s) \text{ for loop} S \\
M_L(STT(s)_k) &= \text{likelihood score for} k \text{ sentence}
\end{aligned}
$$

This says how "probable" it is that the recognized sentence could be an accurate sentence in a particular language.

Lastly, the M_{TTS} model synthesizes the previously recognized sentence $STT(s)_k$ into speech form. At this step, we also calculate the probability for each *K* recognized sentence. The probability gives information on how correctly the M_{TTS} model generates a speech sample for *k* sentence with *s* being the original sample. The score equals:

$$b_k^{lt} = logP(s|STT(s)_k; M_{TTS}) \tag{4}$$

where:

$$
\begin{aligned}
b_k^{lt} &= \text{long-term reward score for} k \\
&\quad \text{sentence of} STT(s) \text{ for loop} S \\
P(s|STT(s)_k; M_{TTS}) &= \text{probability score for receiving} \\
&\quad \text{speech sample} s \text{ from} k \\
&\quad \text{sentence} STT(s), \text{ when synthesized} \\
&\quad \text{using} M_{TTS}
\end{aligned}
$$

2.5 Making use of calculated scores

One iteration in the learning process contains the single performance of both aforementioned loops. After the iteration is completed, we are left with a pair of scores $\left(a_k^{im}, a_k^{lt}\right)$ for each of the *K* generated speech samples and a pair of scores $\left(b_k^{im}, b_k^{lt}\right)$ for each of the *K* recognized sentences.

The scores are then used in a policy gradient algorithm as immediate rewards (a_k^{im} and b_k^{im}) and long-term rewards (a_k^{lt} and b_k^{lt}). We can set the total reward for the *k* sentence (or sample), as:

$$
\begin{aligned}
&a_k = \alpha a_k^{im} + (1-\alpha) a_k^{lt} \\
&\text{or} \\
&b_k = \alpha b_k^{im} + (1-\alpha) b_k^{lt}
\end{aligned}
\tag{5}
$$

where:

$$
\begin{aligned}
\alpha &= \text{a factor specifying the weight of the immediate} \\
&\quad \text{reward in our DSL approach}
\end{aligned}
$$

Having done that, we can formulate the problem as optimizing the a_k and b_k functions. As described before, we

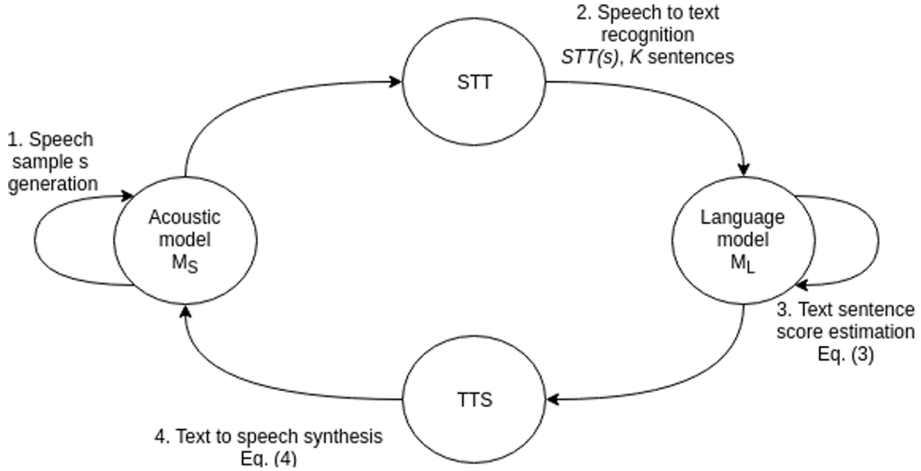

Fig. 2 Second kind of loop. Loop starts from generating a speech sample (loop *S*)

will optimize this function by modifying the weights of two trainable models M_{STT} and M_{TTS}. We use gradient-based methods of optimization. We can calculate gradients of the estimator of the total reward's expected value, with respect to those models. In Eqs. (6) and (7) we depict the calculation for loop L (loop starting from the language model). The calculations for loop S are analogical.

$$\begin{aligned} &\nabla_{M_{TTS}} E[a_k] = \\ &E\left[a_k \nabla_{M_{TTS}} \log P\left(TTS(t)_k | t; M_{TTS}\right)\right] \end{aligned} \tag{6}$$

$$\begin{aligned} &\nabla_{M_{STT}} E[a_k] = \\ &E\left[(1-\alpha) \nabla_{M_{STT}} \log P\left(t | TTS(t)_k; M_{STT}\right)\right] \end{aligned}$$

$$\begin{aligned} &\nabla_{M_{TTS}} \hat{E}[a] = \\ &\frac{1}{K} \sum_{k=1}^{K} \left[a_k \nabla_{M_{TTS}} \log P\left(TTS(t)_k | t; M_{TTS}\right)\right] \end{aligned} \tag{7}$$

$$\begin{aligned} &\nabla_{M_{STT}} \hat{E}[a] = \\ &\frac{1}{K} \sum_{k=1}^{K} \left[(1-\alpha) \nabla_{M_{STT}} \log P\left(t | TTS(t)_k; M_{STT}\right)\right] \end{aligned}$$

where:

$E[a_k]$ = expected reward for ak sample
$\nabla_{M_{TTS}} E[a_k]$ = gradient of the expected reward per$k$ sample, with respect toM_{TTS} model
$\nabla_{M_{STT}} E[a_k]$ = gradient of the expected reward per$k$ sample, with respect toM_{STT} model
$\nabla_{M_{TTS}} \hat{E}[a]$ = gradient of the expected reward, with respect toM_{TTS} model
$\nabla_{M_{STT}} \hat{E}[a]$ = gradient of the expected reward, with respect toM_{STT} model

After calculating the gradients, we can update models M_{STT} and M_{TTS} according to the following formulas:

$$\begin{aligned} M_{TTS} &= M_{TTS} + \eta_{TTS} \nabla_{M_{TTS}} \hat{E}[a] \\ M_{STT} &= M_{STT} + \eta_{STT} \nabla_{M_{STT}} \hat{E}[a] \end{aligned} \tag{8}$$

$$\begin{aligned} M_{TTS} &= M_{TTS} + \eta_{TTS} \nabla_{M_{TTS}} \hat{E}[b] \\ M_{STT} &= M_{STT} + \eta_{STT} \nabla_{M_{STT}} \hat{E}[b] \end{aligned} \tag{9}$$

where:

η_{TTS} = learning rate forM_{TTS} model
η_{STT} = learning rate forM_{STT} model

After one iteration is complete, we start another one, containing both types of loops, and starting from M_L and M_S generating different samples from learned distribution. The proposed DSL process is depicted in Fig. 3.

In this feedback loop setup, both M_{TTS} and M_{STT} models are trained. For our purpose of non-native speech recognition, we pay most attention to M_{STT} and its accuracy. After the training process, the speech recognition model M_{STT}, adjusted to pronunciation features of particular non-native speakers, will be created. Also, M_{TTS} as a speech synthesizer becomes a by-product of the training process. It produces speech biased to the pronunciation patterns of non-native speakers of the language that was in the training dataset.

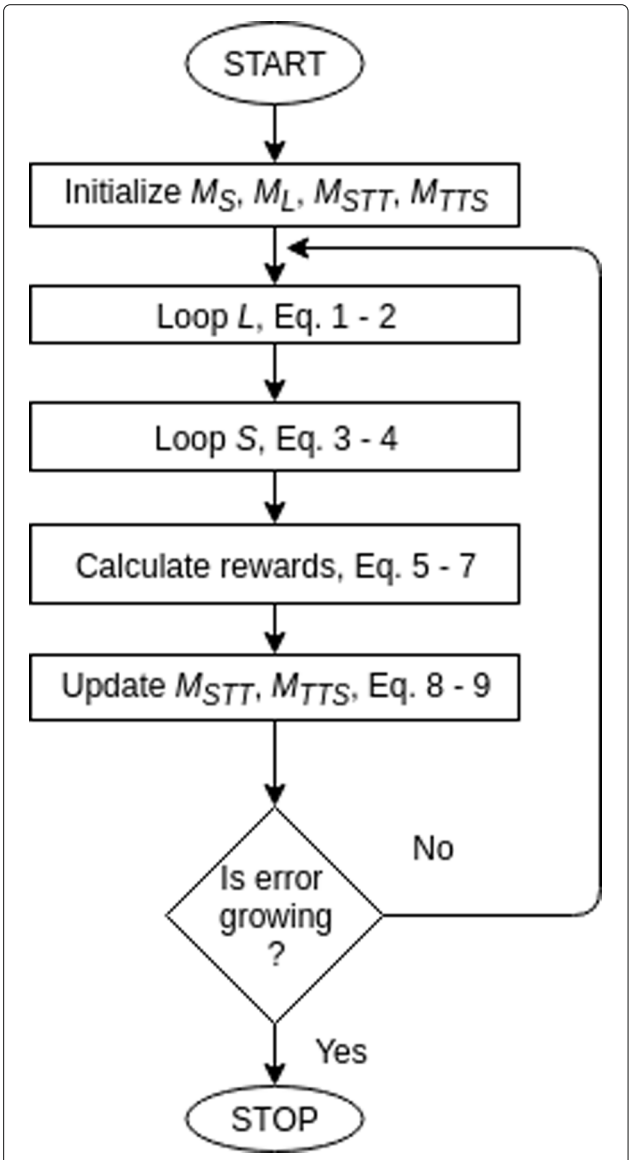

Fig. 3 DSL process for non-native speech recognition

2.6 Experiment setup

2.6.1 Algorithms chosen and tested for each model

We decided to choose several algorithms for each model and test how well the DSL methodology acts in different setups.

Language models M_L in our approach [13]:

- Vanilla recurrent neural network (RNN)
- RNN with a long short-term memory (LSTM) cell
- 3-gram model

The RNN and LSTM language models were created on a character level. A single one-hot encoded row of data which was fed to the network during training was related to one particular character. Then again, during inference time, the network was also fed data samples related to one character. On the other hand, the 3-gram model operates on trigrams of consecutive characters. One aspect of our research was testing whether DSL methodology could be applied and actually useful in different kinds of setups, with different kinds of architectures for each model. That was why we decided to use a 3-gram-based language model in one of our experiments [14, 15].

For acoustic model M_S, we chose the following models [16–18]:

- Vanilla RNN
- RNN with an LSTM cell

The speech recognition models M_{STT} which we decided to examine are as follows:

- Vanilla RNN
- RNN with an LSTM cell

We decided to examine only Deepmind's Wavenet as speech synthesis model M_{TTS}, because speech synthesis was not a primary issue we tried to address in our research.

Table 1 depicts an overview of the architecture of the models.

We tested three different setups of the above models. The architecture of the models in these setups was chosen using local search algorithms in isolated tasks of language modeling, acoustic modeling, and speech recognition.

The first setup (setup 1) contained a 3-layer vanilla RNN with 512 hidden units per layer, for the language model. The same network was used for the acoustic model. As per M_{STT} model, we decided to choose a 2-layer RNN, with 1024 hidden units per layer. Descriptions for setup 2 and setup 3 are analogical to setup 1 and are shown in Table 1.

The reason for choosing the RNN-based neural networks (vanilla RNN and RNN with an LSTM cell) is their performance results on the type of datasets being used in this research. The datasets represented by textual and acoustic domains contain sequences of interdependent data samples. The letters (or words) in any textual sentence that belongs to any text corpus are not to be understood as completely independent of each other. There are sequences of letters where the former ones have a significant impact on which letter may appear as a latter one. Analogical sequential dependency exists in the acoustic domain. An RNN is a straightforward adaptation of the standard feed-forward neural network to allow it to model sequential data. At each timestep, the RNN receives an input, updates its hidden state, and makes a prediction. The RNN's high-dimensional hidden state and nonlinear evolution enable the hidden state of the RNN to integrate information over many timesteps and use it to make accurate predictions. Even if the non-linearity used by each unit is quite simple, iterating it over time leads to very rich dynamics. The standard RNN is given a sequence of input vectors, then it computes a sequence of hidden states and a sequence of outputs. A RNN with an LSTM cell addresses the exploding and vanishing gradient problem, therefore making it possible to track long-time dependencies in the sequential data [19–22].

The aforementioned Wavenet model was designed in a similar manner to Deepmind's original Wavenet [23]. The general idea of the model is to predict the audio sample based on the series of previous audio samples [24]. In order to realize the actual functionality of *TTS*, following the authors' method, we added the possibility to condition the model's prediction locally, on the textual sentence corresponding to a speech sample. In our experiments, we decided to use the Wavenet that consists of three stacks of dilated layers (10 layers per stack, dilation rate up to 512) and two fully connected layers. Other parameters included a filter width of 2, 32 residual channels, 32 dilation channels, and 256 quantization channels.

2.6.2 Types of experiment performed

The purpose of the conducted experiments was to confirm the hypothesis described in Section 2.1 as well as to estimate the accuracy of this method on different setups. In order to assess the quality of this methodology, we designed several experiments (Tables 2, 3, 4, 5, and 6).

Table 1 Model architecture for each setup

Setup	M_L	M_S	M_{STT}	M_{TTS}
1	RNN 3 × 512	RNN 3 × 512	RNN 2 × 1024	Wavenet
2	LSTM 3 × 512	LSTM 3 × 512	LSTM 2 × 1024	Wavenet
3	3-gram	LSTM 3 × 512	LSTM 2 × 1024	Wavenet

Table 2 Results of conducted experiments for setup 1

	English by Japanese	English by Polish
Warm-start	84.12%	83.23%
Semi warm-start	82.92%	81.04%

Table 3 Results of conducted experiments for setup 2

	English by Japanese	English by Polish
Warm-start	89.43%	88.14%
Semi warm-start	88.21%	86.92%

In the first experiment, we decided to check and evaluate the influence of a warm start on the overall accuracy of M_{STT} model. Warm start refers to a training mode where M_{STT} and M_{TTS} models are initially trained with a small amount of labeled data, before we start to train them in a dual supervised manner.

In the second experiment, we checked a semi warm-start approach, training only M_{STT} model with a small amount of labeled data before switching to DSL.

These two experiments were conducted for each of the three model setups.

The last experiment we conducted became a baseline method in our research. This baseline experiment does not make any use of the method we present in this research but instead uses the traditional supervised learning approach, where there is only one, fully labeled dataset.

In this case, we trained a 2-layer RNN with 1024 hidden units in a LSTM cell as M_{STT} model, in a traditional way. In this approach, we trained an end-to-end speech recognition setup that consisted of one network performing conversion from the acoustic domain to the textual one. Because we decided to use the end-to-end model, we used a Connectionist Temporal Classification (CTC) loss function. This loss function does not require a frame-level alignment (matching each input frame to the output token). Therefore, it allows the use of the labeled speech datasets, without the need to align the text with the soundwave frames [25–31].

There was only one model (M_{STT}) in the whole setup, and it was trained in the experiment. We performed the training in a normal, supervised manner, using only a labeled dataset, so that we can show that the results of this traditional approach and the DSL-based one (from previous experiments) are actually comparable.

2.6.3 Datasets used in the experiment

We conducted the first two abovementioned experiments on two cases of Japanese and Polish people pronouncing English sentences.

Table 4 Results of conducted experiments for setup 3

	English by Japanese	English by Polish
Warm-start	86.43%	84.51%
Semi warm-start	85.21%	83.92%

Table 5 Results of conducted experiments for the traditional method (baseline)

	English by Japanese
Traditional method	87.24%

For training language models M_L, we used the Corpus of Contemporary American English (COCA).

For training acoustic models M_S, we used pieces of recordings scraped from Youtube website resources (mostly either Japanese people teaching Japanese to an English audience, or Japanese expatriates living abroad and creating videos in English). The same source was used in the case of Polish people pronouncing English sentences.

During the warm-start and semi warm-start approach, for training M_{STT} and M_{TTS} models, we used 10% (around 7000 recordings) of the *English Speech Database Read by Japanese Students (UME-ERJ)* for Japanese people, and 10% (a similar quantity) of recordings scraped from Youtube for Polish speakers, which we labeled ourselves. The rest of the data was used for verification.

In the last, baseline experiment, we used only the *UME-ERJ* dataset since the amount of time necessary to label the whole scraped dataset for Polish people pronouncing English was too long. In this case, we used 80% (around 56,000 recordings), 10% (around 7000), and 10% of data as training, validation, and testing sets respectively.

A random shuffle strategy was used for selecting each subset of training, testing, and validation sets.

2.6.4 Evaluation of DSL method accuracy

As measure of error, we chose character error rate, or length normalized character-level edit distance. Accuracy is obviously 1 − error.

Since there are not many popular benchmarks for ASR of either Japanese or Polish pronunciation of English sentences, we decided to evaluate the DSL approach for the speech recognition problem by comparing the accuracy result of M_{STT} model created using the methodology described in our paper (DSL) to the accuracy of M_{STT} created using the traditional approach, based on supervised learning (the last of the conducted experiments). In this way, we show that the result yielded by the DSL methodology is comparable to the one achieved by the traditional method. Having said that, we state that the result achieved in the last experiment becomes a baseline result,

Table 6 Average time necessary for training each setup

	Setup 1	Setup 2	Setup 3	Baseline setup
Time	5 weeks	5 weeks	5 weeks	4 days
Epochs	3000	2800	3200	380

against which we compare the results from the first two experiments.

3 Results

The results of our experiments are presented in the table below in Tables 2, 3, 4, 5 and 6. The scores show the best accuracy of M_{STT} model that we managed to obtain during the training process. In order to make the results more reliable, each of the scores shown in the table is an averaged score of six runs of any particular setup. As stated before, during each run, the datasets used for training, testing, and validation were chosen using the random shuffle strategy.

Below, we present how the error rates changed during the training time for the warm-start approach (Fig. 4) with setup 2 and the traditional approach (Fig. 5) for English pronounced by Japanese people.

The warm-start approach chart clearly reflects the moment when we switch from (initial) pre-training to DSL (around the 130th epoch). The convergence rate for the M_{STT} model declines from that point. That means more time is required to achieve comparable results. However, the final accuracy achieved by the warm-start DSL approach is higher.

Even though the DSL method yields better results, they are achieved at a cost of training time. On average, a single run of an experiment using the traditional method took us 4 days to complete using a single GTX 1080 Ti graphics card. The average time needed for a single run of the DSL-based approach to finish was 5 weeks. However, the use of multiple cards allowed us to run the experiments in parallel, and, consequently, to save time. Below, we depict the average necessary time, together with the number of epochs it took to achieve the best result.

While the time necessary for the DSL-based method to achieve the desired results is clearly much longer, it is still acceptable for the purpose of running the experiments and evaluating the methodology.

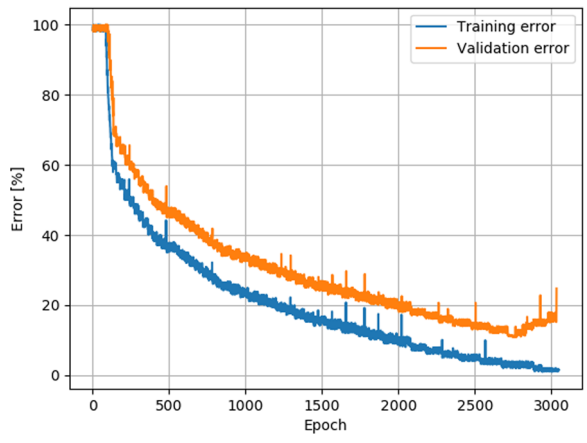

Fig. 4 Training process of warm-start approach. Error rate during the warm-start approach training

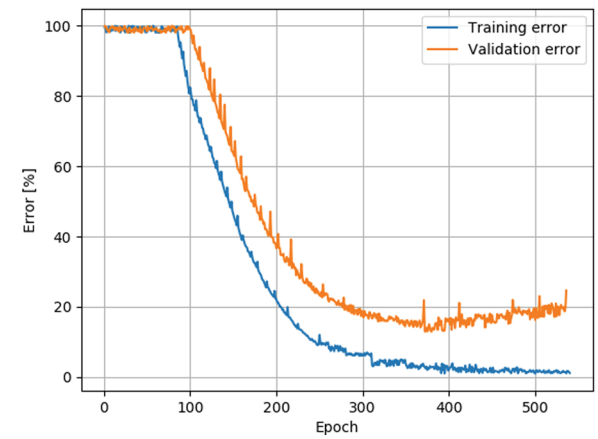

Fig. 5 Training process of traditional approach. Error rate during the traditional approach training

4 Discussion

4.1 Convergence point

Training two networks in such a way that both models learn from one another can bring the risk of the models converging to a point that is not desired. For instance, in the speech recognition and speech synthesis domain, we used M_{STT} and M_{TTS} models. There is a possibility that M_{TTS} may learn pronunciation of a different w word (or sentence), while the language model M_L comes up with a completely different t word. Yet, the immediate reward associated with $M_S(TTS(t))$ may be actually significant since the pronunciation itself is correct according to the acoustic model. If this happens, there is a risk of the M_{STT} model learning to associate the pronunciation of w word with a textual form of t word. The learning process will try to maximize the long time reward associated with $logP(t|TTS(t); M_{STT})$, and in such an event, the M_{STT} model understands that t word becomes a label for an incorrect $TTS(t)$ speech sample (which was mistakenly generated by M_{TTS} earlier). This may lead to a situation where both models learn the incorrect association between speech features and text sentences. Particularly, M_{TTS} can learn the incorrect distribution of $P(TTS(t)|t)$ (i.e., it can learn distribution which would normally represent w sentence). A similar situation may occur for M_{STT} model.

4.1.1 *Warm start and its influence*

Pre-training, or the warm-start approach in a chosen methodology, is helpful for preventing models from learning incorrect associations between speech features and text sentences. It is very useful for speeding up the learning process and increases the chance

of achieving a desired convergence point as it provides a good starting position for the optimization algorithm. Due to the application of pre-trained M_{STT} and M_{TTS} models, we start the DSL process from the point where distributions of $P(TTS(t)|t)$ and $P(STT(s)|s)$ are partially learned from the labeled dataset. Assuming the correctness of the dataset itself, the distributions are correct, but do not represent the full feature space yet.

As we shift from pre-training using labeled datasets into DSL, M_{STT} and M_{TTS} models could expand previously learned distribution using unlabeled data while the learning process continues.

This allows us to both make use of a vast amount of unlabeled datasets and make sure the models are converging towards a desired direction.

4.1.2 Warm start with only one of two pre-trained models

According to the results of our experiments, it appears that the warm start with both models initially trained is not a prerequisite for the models to be correctly trained. One pre-trained model is enough for the whole setup to achieve a desired convergence point.

5 Conclusions

In this research, we explained the problem of non-native speech recognition and the issues that appear if we decide to use traditional approaches for building ASR systems for such cases.

We also described in detail the idea behind DSL methodology and explained why this method is suitable for solving this problem.

Then, we performed experiments, employing different algorithms in different setups, in order to show that DSL methodology can produce ASR systems with an accuracy comparable to currently used ASR products, while at the same time making use of far cheaper and larger unlabeled datasets.

We tested warm-start and semi warm-start approaches, and the results of experiments show that they work well. However, until we have developed the solution to the non-native speech recognition problem in a fully unsupervised manner (without warm start), there is still room for improvement.

Abbreviations

ASR: Automatic speech recognition; DSL: Dual supervised learning; LSTM: Long short-term memory (network); M_{STT}: Speech recognition model; M_{TTS}: Speech synthesis model; M_L: Language model; M_S: Acoustic model; RNN: Recurrent neural network; STT: Speech recognition, speech to text; $STT(s)_k$: k sentence recognized from s speech sample; TTS: Speech synthesis, text to speech; $TTS(t)_k$: k speech sample synthesized from t sentence

Acknowledgements

In our research, we are using a English Speech Database Read by Japanese Students (UME-ERJ), which was provided by Speech Resources Consortium at National Institute of Informatics (NII-SRC) in Tokyo.

Funding

No founding sources were obtained for the purpose of this research.

Availability of data and materials

Please contact author for data requests.

Authors' contributions

KR designed the feedback loop from Section 2.4 and the algorithms mentioned in Section 2.6.1. LW helped with data collecting and labeling scrapped samples. RN and OY provided helpful advice and support during the time of designing the algorithms and experiments. All authors equally contributed to the research and this paper. All authors read and approved the final manuscript.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Author details

[1]Warsaw University of Technology, Institute of Computer Science, Warsaw, Poland. [2]Graduate School of Information, Production and Systems, Waseda University, Kitakyushu, Japan.

Received: 25 June 2018 Accepted: 18 December 2018
Published online: 14 January 2019

References

1. W. Xiong, L. Wu, J. Droppo, X. Huang, A. Stolcke, in *The microsoft 2017 conversational speech recognition system*. Proc. IEEE ICASSP (IEEE, Calgary, 2018), pp. 5934–5938. https://ieeexplore.ieee.org/abstract/document/8461870
2. N. Dave, Feature extraction methods lpc, plp and mfcc in speech recognition. International Journal of Advanced Research Engineering and Technology. **1**, 1–5 (2013)
3. N. Dehak, D. R. P. J. Kenny, P. O. P. Dumouchel, Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech and Language Proceedings. **19**(4), 788–798 (2011)
4. M. Li, N. S. K. J. Han, Automatic speaker age and gender recognition using acoustic and prosodic level information fusion. Computer Speech Lang. **27**(1, January 2013), 151–67 (2013). https://www.sciencedirect.com/science/article/pii/S0885230812000101
5. T. T. D. Drugman, *Glottal closure and opening instant detection from speech signals*. (International Speech Communication Association (ISCA), Brighton, 2009). https://www.isca-speech.org/archive/interspeech_2009/i09_2891.html
6. G. P. A. Shi, M. Shanechi, On the importance of phase in human speech recognition. Audio, Speech, and Language Processing, IEEE Transactions. **14**(5) (2006). Published in: IEEE Transactions on Audio, Speech, and Language Processing. https://ieeexplore.ieee.org/abstract/document/1678004
7. L. M. Tomokiyo, *Recognizing non-native speech: Characterizing and adapting to non-native usage in lvcsr. PhD thesis, Carnegie Mellon University*, (2001)
8. T. P. Tan, *Automatic speech recognition for non-native speakers. PhD thesis*. (Université Joseph-Fourier, Grenoble, 2008). https://hal.inria.fr/tel-00294973/
9. R. Kacper, W. Le, Y. Osamu, in *Non-native english speaker's speech correction, based on domain focused document*. Proceedings of the Conference of Institute of Electrical Engineers of Japan, Electronics and Information Systems Division (The Institute of Electrical Engineers of

Japan, Kobe, 2016). https://ieej.ixsq.nii.ac.jp/ej/?active_action=repository_view_main_item_detail&page_id=13&block_id=18&item_id=88519&item_no=1
10. R. Kacper, W. Le, Y. Osamu, in *Non-native english speakers' speech correction, based on domain focused document*. Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services, iiWAS (ACM, New York, 2016), pp. 276–281
11. R. Kacper, Y. O. W. Le, in *Proceedings of the conference of institute of electrical engineers of japan, electronics and information systems division*. Non-native speech recognition using characteristic speech features, with respect to nationality (The Institute of Electrical Engineers of Japan, Takamatsu, 2017). https://ieej.ixsq.nii.ac.jp/ej/?action=pages_view_main&active_action=repository_view_main_item_detail&item_id=97730&item_no=1&page_id=13&block_id=18
12. Y. Xia, T. Qin, W. Chen, J. Bian, N. Yu, T.-Y. Liu, in *Proceedings of Machine Learning Research. vol. 70, Dual supervised learning*. Proceedings of the 34th International Conference on Machine Learning (International Convention Centre, Sydney, 2017), pp. 3789–3798
13. K. Livescu, J. Glass, in *IEEE International Conference on Acoustics, Speech and Signal Processing*. Lexical modeling of non-native speech for automatic speech recognition. Published in: 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100). (IEEE, Istanbul, 2000). https://ieeexplore.ieee.org/abstract/document/862074. https://doi.org/10.1109/ICASSP.2007.367243
14. F. Bimbot Dept. Signal, P.F..R.P..E.L..B.A. ENST: Variable-length sequence modeling: multigrams. IEEE Signal Processing Letters. IEEE Signal Proc Lett. **2**(6, June 1995), 111–13 (1995). https://ieeexplore.ieee.org/abstract/document/388911
15. S. Deligne Telecom Paris, in *Language modeling by variable length sequences: theoretical formulation and evaluation of multigrams*. FFB., in International Conference on Acoustics, Speech, and Signal Processing (IEEE Conference, Detroit, 1995). https://ieeexplore.ieee.org/abstract/document/479391
16. T. Tan, L. Besacier, Acoustic model interpolation for non-native speech recognition. Proceedings on ICASSP. Published in: 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100). (IEEE, Honolulu, 2007)
17. G. E. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, et al., N.J.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Processing Magazine. **29**(6) (2012). https://ieeexplore.ieee.org/abstract/document/6296526. http://dx.doi.org/10.1109/MSP.2012.2205597
18. T.G.J., Z.Z., F.W., B.S., G.R., in *Proceedings in Interspeech*. Robust speech recognition using long short-term memory recurrent neural networks for hybrid acoustic modelling (International Speech Communication Association (ISCA), Singapore, 2014). https://www.isca-speech.org/archive/interspeech_2014/i14_0631.html
19. S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural Computer. **9**(8), 1735–1780 (1997)
20. T. Mikolov, M. Karafiát, L. Burget, J. Černocký, S. Khudanpur, in *Recurrent neural network based language model*. Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010) vol. 2010 (International Speech Communication Association (ISCA), Makuhari, Chiba, Japan, 2010), pp. 1045–1048. https://www.isca-speech.org/archive/interspeech_2010/i10_1045.html. http://www.proceedings.com/10100.html
21. I. Sutskever, J. Martens, G. Hinton, in *Generating text with recurrent neural networks*. Proceedings of the 28th International Conference on International Conference on Machine Learning. ICML'11 (Omnipress, USA, 2011), pp. 1017–1024. http://dl.acm.org/citation.cfm?id=3104482.3104610
22. A. Graves. Generating sequences with recurrent neural networks, (2013). https://arxiv.org/abs/1308.0850. (arXiv:1308.0850)
23. A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, in *Wavenet: A generative model for raw audio*. Arxiv, (2016). https://arxiv.org/abs/1609.03499. (arXiv:1609.03499)
24. H. Z. K. Tokuda, Directly modeling speech waveforms by neural networks for statistical parametric speech synthesis. Proceedings ICASSP. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). (IEEE, Brisbane, 2015). https://ieeexplore.ieee.org/abstract/document/7178765
25. X. Liu, Deep Convolutional and LSTM Neural Networks for Acoustic Modelling in Automatic Speech Recognition. http://cs231n.stanford.edu/reports/2017/pdfs/804.pdf. Accessed 20 July 2018
26. W. Song, *End-to-end deep neural network for automatic speech recognition*. Published in: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). (IEEE, Conference, Scottsdale, AZ, USA, 2015). http://cs224d.stanford.edu/reports/SongWilliam.pdf. https://ieeexplore.ieee.org/abstract/document/7404790
27. Y. Miao, M. Gowayyed, F. Metze, in *Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding*. Published in: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU) (IEEE, Scottsdale, 2015). https://arxiv.org/abs/1507.08240
28. A. Graves, N. Jaitly, in *Towards end-to-end speech recognition with recurrent neural networks*. Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32. ICML'14 (JMLR.org, 2014), pp. 1764–1772. http://dl.acm.org/citation.cfm?id=3044805.3045089
29. X. Tian, J. Zhang, Z. Ma, Y. He, J. Wei, P. Wu, W. Situ, S. Li, Y. Zhang, *Arxiv*, (2017). https://arxiv.org/abs/1703.07090. (arXiv:1703.07090)
30. A. Graves, G. H. A. Mohamed, Speech recognition with deep recurrent neural networks. Proc ICASSP IEEE. Published in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. (IEEE Conference, Vancouver, 2013). https://ieeexplore.ieee.org/abstract/document/6638947
31. D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, Z. Zhu, in *Proceedings of The 33rd International Conference on Machine Learning, Vol. 48*. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin (PMLR, 2015), pp. 173-82. http://proceedings.mlr.press/v48/amodei16.html