LEUPHANA
UNIVERSITÄT LÜNEBURG

Automatic Speech Recognition

A general introduction

Eva Tesch, *3038587*
Hsin-Yu Ku, *3038591*
Malte Srocke, *3035566*

A written report for the course
Machine Learning Seminar

taught by
Prof. Dr. Ulf Brefeld

in the Master's Program
Management & Data Science

November 29, 2020

1 Introduction

Speech represents a very essential part of our everyday's communication and consists of a sequence of audio signals. Since it comes naturally to us humans, it has also received increasing attention in the field of artificial intelligence which constitutes the study of computers to mimic intelligent human behaviour (OxfordUniversityPress, 2020). This leads us to Automatic Speech Recognition (ASR) which is "the process of converting a speech signal to a sequence of words (i.e., spoken words to text) by means of an algorithm implemented as a computer program" (Karpagavalli & Chandra, 2016).

There is a range of existent applications for ASR, e.g. virtual assistants such as Siri and Alexa, voicemail transcription and captioning of videos, to name a few. The process behind these applications is a similar one and can be represented as a chain of functions in the background. This architecture will be explained in further detail in the subsequent section. In section 3 and 4 we elaborate on two methods commonly used in ASR. We conclude by listing some difficulties and giving a short outlook.

2 Architecture

The traditional architecture of an ASR program comprises several interconnected steps that we aim to explain within this chapter. To follow along, please feel free to look at figure 1 every time a new step is being introduced (indicated by a line break).

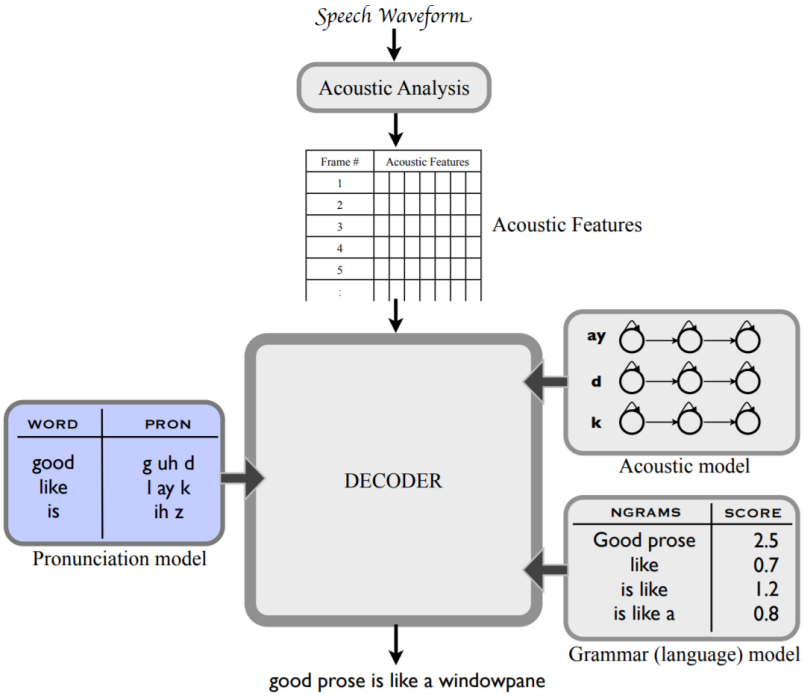

Figure 1: Overview of the process behind ASR (Jyothi, 2019)

Speech recognition starts with the acoustic analysis of speech signals in form of speech waves that are taken as input. The sequence of signals is divided into overlapping slices or frames of about 25ms each, which are supposed to be stationary. For each frame, feature extraction transforms the audio information into a vector of real values. This leads to a sequence of feature vectors as output (Hui, 2019).

As next step the acoustic model converts these features into phonemes, which are basic units of sound. These phonemes are language specific. Most often a word will contain several phonemes in succession. The sound might not be clear however, so that several very similar options of phonemes are stored for later (Sen et al., 2019). In section 3 and 4 we will elaborate further on the acoustic model, explaining two prevalent methods using Hidden Markov Models and Artificial Neural Networks respectively.

The pronunciation model now retrieves these sequences of phonemes and transcribes them into words. Since for one sound several similar phonemes might have been stored in the previous step, this would lead to an output of several potential words here (consider pig vs. pick for example). The transcription does not require a learning phase, but is instead based on a dictionary set up by linguistic experts that includes a list of words and their phonetic translation (Sen et al., 2019).

Having a sequence of word-candidates matching the original audio the language model comes into play. Based on constraints such as statistics and grammar rules, meaningful sentences are derived. This relies on a big training set of text based documents, which serve as an indicator on which pairs, triplets ... of words are more common than others. It is known as the N-Gram model (Jyothi, 2019).

The decoder takes in all previously described models (which are modular and already pretrained), will run through the process and output only the one most probable sentence (or sequence of words) matching the original spoken audio input. Important here is the Viterbi algorithm, for detailed explanation see Sen et al. (2019)

3 Hidden Markov Models (HMMs) in ASR

Until new developments in the recent years, using HMMs in ASR was the standard (O'Shaughnessy, 2008). In some applications and certain forms they are still used today (Karpagavalli & Chandra, 2016).

The main equation of an ASR engine is:

$$\hat{W} = \arg\max_{w}\{P(W \mid X)\} = \arg\max_{w}\{\frac{P(X|W)P(W)}{P(X)}\} \tag{1}$$

where X is the observed audio and the word order is W. The objective is to find the word order W which is most likely for the observed audio X. HMMs are used for the acoustic model, which was mentioned above and is represented by $P(X \mid W)$ (Gales & Young, 2007).

In general, HMMs are used to learn something about unobservable events or states by observing another process which is thought to be dependent on the unobservable (or hidden) states (Jurafsky & Martin, 2014). The elements of an HMM are the states

($S = \{s_0, s_1, ..., s_N\}$), the transition probabilities ($P(q_t = S_i \mid q_{t-1} = S_j) = a_{ji}$), the observations of the observable process ($O = \{o_1, o_2, ..., o_T\}$) and the emission probabilities ($P(y_t = o_k | q_t = s_j) = b_j(k)$).

In the context of speech recognition, the observable process is the speech represented as a sequence of audio frames of 25ms transformed into a feature vector, as explained above. The unobservable states are the words, which consist of phones which again are represented by sub-phones (see figure 2).

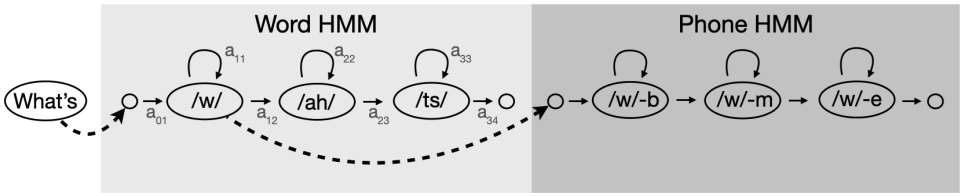

Figure 2: HMM-based word and phone model (Author's own rendering)

As stated above, the basic units of sound are phonemes which make up words. The actual sound heard or recorded is called a phone. Trying to build a model around words directly would not be feasible due to significant differences in pronunciation. Therefore the unit phone is used. The further division into sub-phones is necessary because the acoustics of a phone are affected by the phonemes preceding and succeeding (Pearce et al., 1991).

Language is modeled as being linear in time. As a result, there are only two possible transitions from a word, phone or sub-phone: either staying in itself (in figure 2 that would be a_{11}, a_{22} or a_{33}) or moving to the next word, phone or sub-phone (in figure 2 that would be a_{12}, a_{23} or a_{34}). Going back to a previous state, which usually is possible in an HMM, is not possible in the context of ASR (Rabiner & Juang, 1993).

The emission probabilities (the distribution of features for a phone) can be modeled with a Gaussian Mixture Model (GMM) and learned from training data (Gales & Young, 2007). This combination of Hidden Markov Models with Gaussian Mixture Models is referred to as GMM-HMM.

According to Rabiner and Juang (1993) there are three problems which characterize an HMM: evaluation, decoding and training. The evaluation problem means that one wants to compute the probability that a particular observation sequence was produced by the model. In ASR that would be the probability of a particular sequence of audio frames. To compute this, the Forward algorithm is used, which is a kind of dynamic programming algorithm. Decoding means that one wants to find the state sequence which maximizes the probability of a particular observation sequence. In the case of ASR, this means finding the sequence of phones or sub-phones which fit best to a given sequence of audio frames. To do this, the Viterbi Algorithm is used. Training means that given an observed sequence and the set of possible states in the HMM, the parameters for the HMM should be learned. The algorithm used for this is the forward-backward, or Baum-Welch algorithm.

4 Neural Network for ASR : from partial to end-to-end

HMM marks the era when statistical models were used for ASR starting in 1980s (Karpagavalli & Chandra, 2016). Over decades, GMM-HMM was the most common learning approach before the combination of artificial neural networks (ANNs) and HMMs was applied as an alternative paradigm for ASR in 1990s (Yu & Deng, 2015). However, it was not until deep neural network (DNN) came in the forefront in 2006, that the power of neural network is widely applied in ASR. With its many feed forward hidden layers, DNN used in the hybrid system with HMM (referred to as DNN-HMM) has been proven to outperform GMM-HMM at acoustic modeling on a variety of speech recognition benchmarks (Hinton et al., 2012). One example is the application for large vocabulary continuous speech recognition tasks in Dahl et al. (2012).

4.1 DNN for acoustic model (DNN-HMM)

Under DNN-HMM, GMMs were replaced by DNNs in the acoustic model of ASR to map acoustic features to phonemes. That is, the emission probabilities over HMM states, which is the part of $P(W \mid X)$ in equation 1, is estimated by DNN instead of GMM. The architecture of DNN-HMM can be seen in figure 3.

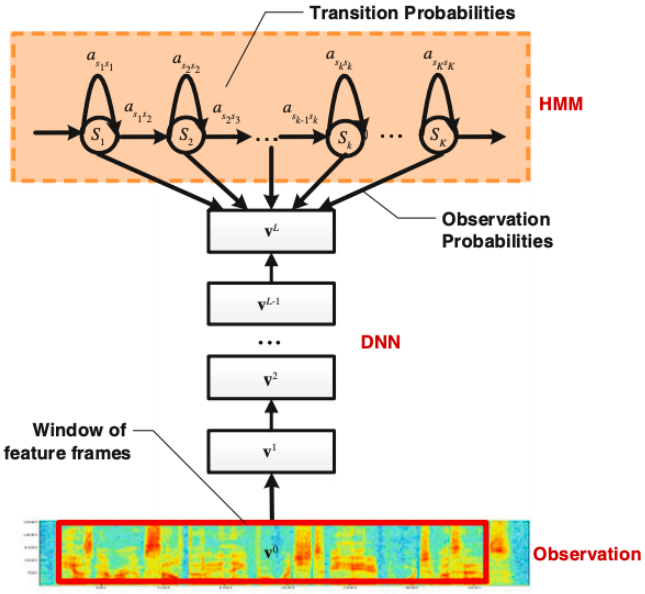

Figure 3: An architecture of the DNN-HMM hybrid system (Yu & Deng, 2015)

For ASR application, DNN has its drawback due to the lack in ability to persist past information. DNN has difficulties to deal with different speaking rates and temporal dependencies in ASR. This rationalizes the introduction of recurrent neural network (RNN) to ASR tasks.

4.2 RNN-LSTM: from acoustic model to end-to-end ASR

RNN is a class of neural network models that contains loops in the hidden layer. If we unroll the loop, RNN can be thought of as multiple copies of the same network, each network passing a message to a successor. This chain-like nature shown in figure 4 shows that RNNs are clearly related to modeling sequence data like text or speech. This structure empowers RNN to handle different speaking rates and temporal dependencies in ASR (Shewalkar et al., 2019).

Mathematically, given an input sequence $x = \{x_0, x_1, ..., x_T\}$, a RNN computes the hidden vector sequence $h = \{h_0, h_1, ..., h_T\}$ and the output vector sequence $y = \{y_0, y_1, ..., y_T\}$ by iterating the following equations from t = 1 to T:

$$h_t = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{2}$$

$$y_t = W_{hy}h_t + b_y \tag{3}$$

W terms denote weight matrices (W_{xh} is the input-hidden weight matrix); b terms denote bias vectors (b_h is the hidden bias).

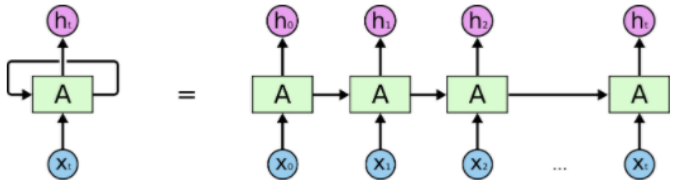

Figure 4: A sketch of a RNN model (Christopher Olah, 2015)

In particular, an advanced type of the RNN called long-short-term memory (LSTM) is designed with a memory cell to deal with long-term dependencies in data. It is proved to be more effective than DNN and conventional RNN for acoustic modeling (Sak et al., 2014). Recent research demonstrates two variations of LSTM for ASR task: deep LSTM-RNN and bidirectional LSTM-RNN. The former stacks multiple RNN hidden layers on top of each other, while the latter makes use of both previous and future context in two directions. A combination of the two, called deep bidirectional LSTM, has been employed not only for the acoustic model (Sak et al., 2014), but also as an end-to-end model for the whole ASR pipeline. This means the model directly learns from acoustic features to phonetic sequences. It can be illustrated by replacing all the parts in figure 1 after acoustic features into a single neural network model. Graves et al. (2013) and Graves and Jaitly (2014) showed that deep bidirectional LSTM with end-to-end training gives state-of-the-art results in ASR tasks.

5 Summary and Outlook

In the first section a high-level overview and introduction to ASR was given, mentioning the most important parts of an ASR engine, namely the feature extraction followed by the decoder taking in results from the acoustic, pronunciation and language model. Then two approaches of implementing those models were explained: the former state-of-the-art HMMs and the different forms of neural networks which are nowadays mostly used and give the best results.

There are several difficulties within setting up an ASR program that should be further addressed in the future. These come from noisy settings, a variability of pronunciation and different accents of a language. Moreover, recognizing a switch of language within an audio input needs further research. Finally, another challenge lies within applying an ASR program to a new language without a lot of additional (manual) work needed. This also includes that the algorithms should be trained better to work on little training data (Microsoft Research, 2017).

References

Christopher Olah. (2015). *Understanding LSTM*. Retrieved November 28, 2020, from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*(1), 30–42.

Gales, M., & Young, S. (2007). The application of hidden markov models in speech recognition. *Foundations and Trends® in Signal Processing*, *1*(3), 195–304.

Graves, A., Mohamed, A., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 6645–6649.

Graves, A., & Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, II–1764–II–1772.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, *29*(6), 82–97.

Hui, J. (2019, December 20). *Speech recognition — phonetics* [Medium]. Retrieved November 28, 2020, from https://jonathan-hui.medium.com/speech-recognition-phonetics-d761ea1710c0

Jurafsky, D., & Martin, J. H. (2014). *Speech and language processing* (2nd ed.) [Series Title: Always learning]. Pearson Education.

Jyothi, P. (2019). *Automatic speech recognition*. https://www.cse.iitb.ac.in/~pjyothi/cs753/slides/lecture1.pdf

Karpagavalli, S., & Chandra, E. (2016). A review on automatic speech recognition architecture and approaches. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, *9*(4), 393–404.

Microsoft Research. (2017, September 11). *Automatic speech recognition - an overview*.

O'Shaughnessy, D. (2008). Invited paper: Automatic speech recognition: History, methods and challenges. *Pattern Recognition*, *41*(10), 2965–2979.

OxfordUniversityPress (Ed.). (2020). Artificial intelligence. In *Oxford advanced learner's dictionary*.

Pearce, D., Wood, L., & Novello, F. (1991). Improved vocabulary-independent sub-word hmm modelling. *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 181–184.

Rabiner, L. R., & Juang, B.-H. (1993). *Fundamentals of speech recognition* [Series Title: Prentice Hall signal processing series]. Prentice Hall PTR.

Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. *INTERSPEECH*.

Sen, S., Dutta, A., & Dey, N. (2019). *Audio processing and speech recognition: Concepts, techniques and research overviews*. Springer Singapore.

Shewalkar, A., Nyavanandi, D., & Ludwig, S. A. (2019). Performance Evaluation of Deep Neural Networks Applied to Speech Recognition: RNN, LSTM and GRU. *Journal of Artificial Intelligence and Soft Computing Research*, *9*(4), 235–245.

Yu, D., & Deng, L. (2015). *Automatic speech recognition*. Springer London.