Analysis of Automatic Speech Recognition Methods

Olena Iosifova[a,b], Ievgen Iosifov[a], Volodymyr Sokolov[b], Oleh Romanovskyi[a], and Igor Sukaylo[b]

[a] *Ender Turing OÜ, 1/2 Padriku str., Tallinn, 11912, Estonia*
[b] *Borys Grinchenko Kyiv University, 18/2 Bulvarno-Kudriavska str., Kyiv, 04053, Ukraine*

Abstract

This paper outlines structures of different automatic speech recognition systems, hybrid and end-to-end, pros and cons for each of them, including the comparison of training data and computational resources requirements. Three main approaches to speech recognition are considered: hybrid Hidden Markov Model – Deep Neural Network, end-to-end Connectionist Temporal Classification and Sequence-to-Sequence. The Listen, Attend, and Spell approach is chosen as an example for the Sequence-to-Sequence model.

Keywords

Automatic speech recognition, ASR, hidden Markov model, HMM, deep neural network, DNN, LAS, hybrid, end-to-end, sequence-to-sequence, speech recognition, speech-to-text.

1. Introduction

Automatic speech recognition (ASR) is a very popular technology that is widely adopted in a real-life environment and business applications. Mobile phones provide speech-to-text functions through a variety of applications, voice assistants route incoming calls, "communicating" with clients that call their bank or insurance company every day, and drivers command their cars with voice. What makes it so widespread is our (human) natural way of communication—speech. We learn to speak quite early and practice it every day unlike the communication with different technologies that we usually do through different user interfaces. All of the interfaces are built differently and it makes communication with new technologies complicated. We have to learn all the time we get something new in a set of our gadgets or apps. Recent studies in voice interfaces give us an idea that in the future the problem of an overwhelming variety of user interfaces will be solved [1, 2]. All of it is possible because of highly accurate state-of-the-art ASR systems [3].

The content of our work has the following structure: Sect. 2 shows the speech recognition approach. In Sect. 3, we review related works and the most prominent directions of research. In Sect. 4 we overview hybrid ASR systems based on Hidden Markov Model (HMM) and Deep Neural Networks (DNN), we also give a conclusion on drawbacks this architecture has but point out also very important strong parts of it. Sect. 5 follows with the observation of end-to-end systems including sequence-to-sequence models. Its' best parts and limitations. The possible future work is described in Sect. 6.

2. Speech Recognition Approach

The high-level idea behind the speech recognition is to convert captured audio signals into relevant textual representation (Fig. 1).

Cybersecurity Providing in Information and Telecommunication Systems, January 28, 2021, Kyiv, Ukraine
EMAIL: oi@enderturing.com (A.1); ei@enderturing.com (A.2); v.sokolov@kubg.edu.ua (B.3); or@enderturing.com (A.4); i.sukailo.asp@kubg.edu.ua (B.5)
ORCID: 0000-0001-6507-0761 (A.1); 0000-0001-6203-9945 (A.2); 0000-0002-9349-7946 (B.3); 0000-0003-3420-5621 (A.4); 0000-0003-1608-3149 (B.5)

Figure 1: Speech recognition schema

The ASR system represents input sequence of sounds (or acoustic input) $X = \{x_1, x_2, \ldots, x_T\}$ of length T as a resulting sequence $Y = \{y_1, y_2, \ldots, y_L\}$ (which can be characters, chunks of words or words) of length L.

Modeling of ASR systems is the task of creating the generative model [4]. The classical way to create ASR is to build a language model, a pronunciation model, and an acoustic model. Just until recently, all three components have been essential for the task of speech recognition (Fig. 2).

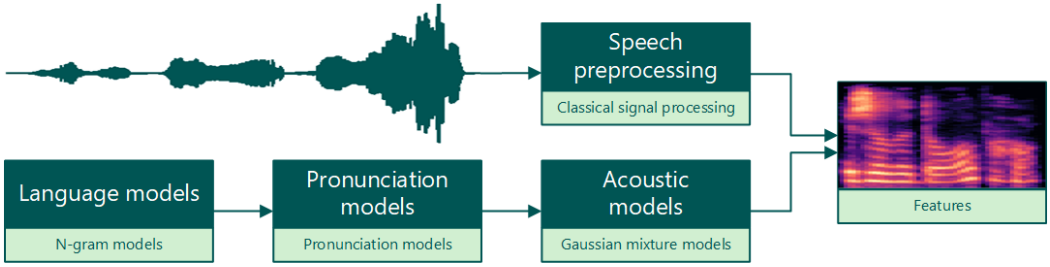

Figure 2: Multi-module automatic speech recognition system modelling

From a language model, we produce a sequence of words. A pronunciation model gives a result of how a particular word is spoken. We would find it is written out as the sequence of phonemes, which are basic units of sound (sequence of tokens). A pronunciation model converts the sequence of text into a sequence of pronunciation tokens. After that, it is fed into an acoustic model, which produces a result of how a given token sounds like.

In this pipeline, each component has its statistical model. The final result of all the models working together is to infer the most probable textual sequence $Y' = \{y_1, y_2, \ldots, y_L\}$ with given data (audio features) $X = \{x_1, x_2, \ldots, x_T\}$

$$Y' = \arg\max_{Y} p(X|Y)\, p(Y). \quad (1)$$

However, the introduction of end-to-end models changed the architecture so there is no need for many separated components.

In our work, we provide an overview of ASR systems that are based on hybrid HMM-DNN architecture and end-to-end models, paying attention to the achieved results in speech recognition accuracy. Specifically, we will analyze the paradigm of the Connectionist Temporal Classification (CTC) model, Listen, Attend, and Spell (LAS) model that is a sequence-to-sequence model, and the most recent architectures of sequence-to-sequence online models. We outline their advantages and drawbacks. Unfortunately, due to the high variety of training approaches, model architectures, languages involved, and training and test data diversity, there is no possibility to gather the objective performance results comparison even from a big number of papers published in this area. Therefore, we will provide examples of the highest currently published results, to the best knowledge of the authors, of different approaches and types of ASR systems that we observe in this work. Finally, we give the conclusion and ideas of future works.

3. Related Works

Despite that fact of great results of hybrid systems and less demand for training data and resources still, we have to admit that future works are mostly focused on online end-to-end systems that will penetrate most of the spheres of ASR application in the future: IoT, voice assistants, human-machine communication, etc. Therefore, we have to pay separate attention to those and explore it in detail in our future work.

Going through several papers devoted to online end-to-end systems we notice that authors focus on solving problems of extensive data requirements, the Word Error Rate (WER) for languages without a high-quality pronunciation lexicon as well as some other narrow tactics for better results [5].

One of the interesting ideas has been proposed by a team of researchers from Facebook and Microsoft. In the work [6], they used a teacher-student learning initialization strategy to transfer knowledge from a sophisticated off-line end-to-end model to an online end-to-end speech recognition model. And this helped them to eliminate the need for a quality lexicon or any other linguistic add-on. This idea has been evaluated on a Microsoft Cortana. This is an NLP task connected to a personal assistant. The method, that the authors proposed method shows a 19% improvement in WER.

What we have found even more interesting is a direction of research to reduce the requirements for training data in online end-to-end systems. The authors of [7] focused their research exactly on this idea. The importance of this direction is explained with practical needs. Specialists in a production environment regularly have tasks to build ASR systems for new downstream tasks with limited domain data and in a short period. End-to-end methods that are currently the most researched area, significantly ease the model's preparation. But nevertheless, it has the training data requirements issue. Authors experiment with a number of techniques for creation of an online Automatic Speech Recognition systems like an end-to-end model but with a much smaller of data related to the specific domain. In particular they use data augmentation for the target domain, fine-tuning of previously pretrained models, that have been trained on a large corpus of data, i.e., transfer learning. And finally, the model distillation for the parameters on the unlabeled data with the help of a 'teacher' - bi-directional model. All of the described cases are possible to use in downstream tasks solutions. We can acknowledge that proposed techniques are independent and applicable for increasing the performance of Automatic Speech Recognition in the desired domain.

4. Hybrid Hidden Markov Model and Neural Networks

HMM suits the modeling of time-varying spectral vector sequences as a very efficient framework [8]. Therefore, most current continuous speech recognition systems are based on HMMs. Most of the first built recognition systems used HMM to model the speech state and Gaussian Mixture Model (GMM) to model HMM states' observation probability. It was considered a breakthrough in speech recognition approaches until neural networks invasion.

In 2011, Microsoft Research presented a hybrid system (CD-DNN-HMM) where the HMM was combined with a context-based DNN [9]. The result was significantly better compared to the HMM-GMM system. In 2012, HMM-DNN largely outperformed state-of-the-art HMM-GMM systems [10].

Generally speaking, HMM-based speech recognition systems contain three parts, as described in Section 1: acoustic, pronunciation, language models. Each of these parts can be built on HMM in combination with neural networks. And having its statistical model each component provides its resulting hypotheses about their results. So altogether it gives the following statement:

$$\arg\max_{Y} Y * p(Y|X) \approx \arg\max_{Y} Y * \sum_{S} S\, p(X|S)p(S|Y)p(Y) \quad (2)$$

where S is HMM state sequence $S = \{s_t \in \{1, 2, \ldots, I\} \mid t = \{1, 2, \ldots, T\}\}$, and $p(X|S)$, $p(S|Y)$, and $p(Y)$ reflect acoustic model, pronunciation model, and language model [2].

Despite the high performance of end-to-end systems, hybrid HMM-DNN and HMM-(B)LSTM (Bidirectional Long Short-Term Memory) systems are dominant in a lot of production environments. Real-life problems have their demands, for example, in the vast majority of cases there is extensively more text data than audio or the real task requires several separate language models, so in these situations HMM-DNN/(B)LSTM is a logical choice [11].

An interesting comparison of end-to-end vs hybrid HMM-DNN was performed by [12]. To compare hybrid HMM-DNN and end-to-end systems following architectures were chosen. The hybrid DNN/HMM and attention-based systems both had BLSTMs for acoustic modeling/encoding. Also, LSTM and Transformer-based architectures were used for language models. The end-to-end system had an attention-based encoder-decoder design. The training set is LibriSpeech. The best WER achieved for both systems were: 8.4% for end-to-end system and 4.5% for hybrid HMM-BLSTM with Transformer-based language model. The described in [12] Hybrid system outperformed the end-to-end

system by 40%. These results had been achieved on 960 hours of training data. But authors also state that with the significant increase in the amount of training data the gap in results of both systems shrinks dramatically. The lower the amount of data, the more performing is hybrid systems.

All of the observed above gives us ground to conclude that hybrid systems possess the following limits:

- Multi-module architecture makes it complex in design, training, and optimization.
- As each of the hybrid system's components has its statistical model and own neural network, it produces independent errors that are not coherent among all of them so it makes it an even more complicated task to achieve great results.
- And the most influencing drawback is that application of neural networks is limited to the function of HMM states observation that gives us an idea of the system to face the ceiling in the development.

However, as was stated above, there still valuable qualities of the hybrid HMM-DNN/(B)LSTM systems:

- State-of-the-art results that outperform end-to-end systems in a variety of tasks.
- Much better results on a much lower volume of training data.
- Possibility to solve efficiently several real-life tasks with lower training and computational resources involved.

5. End-to-End Automatic Speech Recognition

All the limitations of the above-mentioned hybrid multi-module systems became an inspiration for researchers to create a process when the entire model is trained as one single big model, which later got a classification as an end-to-end model. What it does is just digests data, or features $X = \{x_1, x_2, \ldots, x_T\}$ and produces resulting sequence $Y = \{y_1, y_2, \ldots, y_L\}$ with just one powerful probabilistic model $Y = p(Y|X)$.

The first such model is called CTC and was introduced by [13]. And it replaced HMM in a model architecture. The CTC-based models can directly output the final transcripts, while HMM-based models mostly output small units like phonemes or others, and a lot of the following processing is needed to get the results. Applying CTC significantly simplifies the architecture and training of the model. It was a great achievement at a time. Having its significant limitations it produced a further idea or building sequence-to-sequence models [14, 15]. However, CTC-based models still have their place in a production environment, like Google, Baidu, etc., so this is an important point in ASR development and the current technology stack.

5.1. Connectionist Temporal Classification Model

The speech recognition with a CTC-based model goes through two important processes: path probability calculation and path aggregation. Path probability calculation goes as follows. The spectrogram (features X) is fed into a bidirectional Recurrent Neural Network (RNN). The vocabulary for CTC is the labels, it can be letters $\{a, b, c, \ldots, z\}$ and an extra token <b> called a 'blank token'. Each frame of the prediction is a log probability for a different token class the according to time step. It is called a score s. And the complete equation is:

$$s(k,t) = \log P_r(k,t|X) \quad (3)$$

where softmax at step t gives a score $s(k,t)$, and it is a log probability of category k, at a time step t, given the data X. Several results of softmax function are produced by the RNN over the entire step. The result of the model work should be the probability of the transcript through these individual probabilities over time. So, the system can take a path through the entire space of softmax function results, and look at just the symbols that correspond to each of the time steps. Then aggregation comes into place. From the path probability calculation process, it can be figured out that the length of the output path is equal to the length of the input speech sequence, which does not match with the real data. In the majority of cases, the length of transcription is shorter than the length of the input speech sequence. A many-to-one, long-to-short mapping is a necessity to aggregate multiple paths into a shorter label sequence.

5.2. Sequence-to-Sequence Model

Although the CTC model is great, from a modeling perspective, you would find that the model makes predictions just based on the current data. And once it's done with making those predictions for each frame, there's no way of adjusting that prediction. It has to the best it can with those predictions.

An alternative end-to-end architecture that does not need intermediate steps is the sequence-to-sequence model [16] of which the main function is to generate next-step prediction at any arbitrary point of time, using all previous data. The main limitation of sequence-to-sequence implementation for speech recognition is the ability to track long sequence dependencies, and if in the text we talk about 10–20 time steps, audio sequence, on the other hand, is much longer, and dependencies have to be tracked at a distance of around hundreds of time steps. This limitation was partially solved by the attention vector mechanism and by the hierarchical encoder mechanism, which looks in a very narrow interval around the current timestamp to build an attention vector.

For instance, LAS is one of the sequence-to-sequence implementations for audio sequence. LAS produces multiple outputs with probabilities for input sequence (multimodal outputs). And this is why this one model can learn such complex functions, because the more mistaken outputs it produces, the more feedback it has. Moreover, the model can learn very specific train dataset patterns, which makes this model a good candidate for domain fine-tuning. Another strong part of LAS is causality, which means the model can predict for example digit instead of the word representation. LAS still can benefit from external Language model—it is no substitute for billions of words texts for language model, so it is still a better idea to have two datasets, one for the ASR model, and another one to train language model to use as an additional layer on top of the ASR model.

There are few limitations in sequence-to-sequence models for speech:

- As it is an encoder-decoder-based model, it is not an online model, which means it is the next token prediction and we have to have the full sequence to produce output, we cannot give just chunk by chunk like in Hybrid models.
- As a result of the previous, we cannot generate accurate start and end of words using the sequence-to-sequence model.
- Attention mechanisms is a computational bottleneck for audio sequence.
- Accuracy is much lower for short sequences, which makes current architecture hard to implement for speech conversational systems, where customers can utterance just "yes" as an answer to system questions.

6. Conclusions and Future Work

In this paper, we reviewed the most prominent and actual speech recognition approaches and architectures [1, 12–14]. While most of the production-based implementations still rely on HMM-DNN architecture which requires a lot less dataset to be trained we see a strong trend towards end-to-end approaches which doesn't require intermediate steps and hence engineering skills. Briefly comparing automatic speech recognition models, we can say that the most complex in building remain hybrid systems and its' representative in our work, HMM-DNN, while it provides the best results among others on a number of downstream tasks. The next by complexity goes CTC-based model. Despite the fact it is an end-to-end model, it is unable to learn language model so to get relevant WER, engineers have to link it with a separate language model that makes it inconvenient, complex, less performing. The easiest to build and maintain are end-to-end sequence-to-sequence models, like LAS or RNN Transducer (RNN-T), but currently for many real production tasks these models don't show their applicability.

The current main limitations of end-to-end architectures are the amount of data and computational resources to achieve comparatively to HMM-DNN accuracy. It just requires a few times more input datasets and more than 10 times more computational resources. At the same time, we believe in the future end-to-end architectures will become more efficient and advanced which leads to broader production implementation of end-to-end speech recognition approaches. Such an approach significantly decreases required skills and will most probably make speech recognition more affordable do develop.

In the future, we will focus on the direct comparison of three architectures in terms of accuracy using the same input datasets and will continue to follow end-to-end improvements to catch the moment when they will be ready to provide comparative accuracy while providing comparative effectiveness.

7. Acknowledgments

The research team is grateful to Ender Turing OÜ for defining the business problem, comments, corrections, inspiration, and computational resources.

8. References

[1] T. Tanaka, et al., A joint end-to-end and DNN-HMM hybrid automatic speech recognition system with transferring sharable knowledge, in: Interspeech, 2019, 2210–2214. doi:10.21437/interspeech.2019-2263.
[2] D. Wang, X. Wang, S. Lv, An overview of end-to-end automatic speech recognition, Symmetry, 11(8) (2019) 1–26. doi:10.3390/sym11081018.
[3] 14 I. Iosifov, O. Iosifova, V. Sokolov, Sentence segmentation from unformatted text using language modeling and sequence labeling approaches, in: Proceedings of the 6th International Scientific and Practical Conference Problems of Infocommunications. Science and Technology, October 6–9, 2020, pp. 1–4. To appear.
[4] E. McDermott, A deep generative acoustic model for compositional automatic speech recognition, in: 32nd Conference on Neural Information Processing Systems, Montreal, 2018, pp. 1–17.
[5] S. Zhang, et al., Streaming chunk-aware multihead attention for online end-to-end speech recognition, 2020, pp. 1–5. arxiv:2006.01712. To appear.
[6] S. Kim, et al., Improved training for online end-to-end speech recognition systems, in: Interspeech 2018, pp. 2913–2917. doi:10.21437/interspeech.2018-2517.
[7] Y. Chen, Data techniques for online end-to-end speech recognition, 2020, pp. 1–5. arxiv:2001.09221. To appear.
[8] L. E. Baum, J. A. Eagon, An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology, Bull. Am. Math. Soc. 73(3) (1967) 360–364. doi:10.1090/s0002-9904-1967-11751-8.
[9] G. E. Dahl, et al., Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition, IEEE Trans. Audio Speech Lang. Process. 20(1) (2012) 30–42. doi:10.1109/tasl.2011.2134090.
[10] G. Hinton, et al., Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, IEEE Signal Process. Mag. 29(6) (2012) 82–97. doi:10.1109/msp.2012.2205597.
[11] O. Romanovskyi, et al., Automated pipeline for training dataset creation from unlabeled audios for automatic speech recognition, in: Proceedings of the 3rd International Conference on Computer Science, Engineering and Education Applications, 2020, pp. 1–12. To appear.
[12] C. Luscher, et al. RWTH ASR systems for LibriSpeech: Hybrid vs attention—w/o data augmentation, 2019, pp. 1–5. arxiv:1905.03072. To appear.
[13] A. Graves, et al., Connectionist temporal classification, in: Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 369–376. doi:10.1145/1143844.1143891.
[14] R. Hsiao, Online automatic speech recognition with listen, attend and spell model, 2020, pp. 1–5. arxiv:2008.05514. To appear.
[15] W. Chan, et al., Listen, attend and spell: A neural network for large vocabulary conversational speech recognition, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2016, pp. 1–16. doi:10.1109/icassp.2016.7472621.
[16] O. Iosifova, et al., Techniques comparison for natural language processing, in: Proceedings of the 2nd International Workshop on Modern Machine Learning Technologies and Data Science, June 2–3, 2020: no. I, vol. 2631, 2020, pp. 57–67.