Tackling Data Sparsity and Domain-Specific Accented Speech: Improving an ASR System using the Kaldi Toolkit

Vanessa Dengel, Tuo Zhang

Institute for Natural Language Processing
University of Stuttgart
Team Lab Phonetics SS 2021
Supervisor: Pavel Denisov

vanessa.dengel@ims.uni-stuttgart.de, tuo.zhang@ims.uni-stuttgart.de

Abstract

In a traditional ASR system the language model, pronunciation dictionary and acoustic model are crucial components driving the recognition quality. In this project, it is shown that creating and interpolating domain-relevant language models improves the word error rate significantly. In contrast to that, replacing a speaker-specific acoustic model, even though built on sparse data, with a more general one does not lead to further improvement, but to a performance drop. In addition, adding domain-specific vocabulary to the pronunciation dictionary and modifying the pronunciation dictionary according to the accent of the speaker improves the performance of the ASR system as expected.

1. Introduction

1.1. Project Objective

The goal of this project is to improve an automatic speech recognition (ASR) system by optimizing the language model, pronunciation dictionary and acoustic model using the kaldi toolkit. Related to that is the motivation to test and compare different approaches for modifying the respective components and the resulting outcomes. The baseline system is built with the mini librispeech recipe on sparse acoustic data of accented speech (198 utterances), namely the training data of Pavel's kaldi tutorial. The model is evaluated and tuned on a distinct development set of the same speaker containing a total of 119 utterances. For this baseline, the best achieved word error rate (WER) is 67.1% for the monophone model and 64.93% for the triphone model. Dealing with data sparsity and domain-specific accented speech are the main challenges of this project.

1.2. Kaldi toolkit

Kaldi is an open-source toolkit for ASR research. The toolkit provides an ASR system based on finite-state transducers, documentation and scripts for building recognition systems. It is intended to make easily extensible and modifiable code available to researchers. The core library supports acoustic modeling with subspace Gaussian mixture models as well as standard Gaussian mixture models [1].

2. Related Work

Language modeling takes an important part in speech technologies like speech recognition systems [2]. Here, n-gram language models have proven to be a simple but effective method [3]. Broman and Kurimo [2005] combined such language models in different ways, namely (log) linear interpolation, bin estimation and unigram rescaling, to compare the outcome, advantages and drawbacks of the respective methods. Their results suggest that all of these language model combination methods improve the performance compared to a four-gram model baseline. The best result is achieved with the bin estimation method. However, large amounts of data are needed here and so it is not generally suitable. In addition, they find linear interpolation as a popular method achieving good results that enables fast calculation due to an easy parameter estimation and the avoidance of normalization.

Regarding acoustic modeling, Hunter [2004] finds that speaker independent systems tend to show a weaker performance than a comparable speaker dependent one [4]. The author sees the reason for that in the variability existing among speakers which a system can hardly capture adequately.

As we want to develop a pronunciation dictionary which adapts to the speaker's accent, we have consulted [5] and [6], which have described some of the accentual features of English speakers from Russian background.

3. Data & Methods

3.1. Language Model

In order to find a good language model, parameters are tuned and more domain specific data is added. The created and applied language models are n-gram models that determine the next word based on the probability of a word *n* following the preceding *n-1* words.

To deal with the challenge of data sparsity and domain-specific speech, we selected the TED-LIUM corpus in addition to the training data of Pavel's kaldi tutorial. TED-LIUM contains transcripts of scientific talks from different domains, summing up to a total of 183710 utterances [7]. On the one hand, this corpus might provide relevant data that helps the language model to generalize better and on the other hand keep focusing on domain-specific expressions. Since we aim to build an ASR system for scientific but still rather spontaneous spoken language, it is further important to take into account the differences between written and spoken speech. This aspect is supposed to be captured by the choice of grounding the language modeling on these data.

In addition, we created a small corpus of the kaldi for dummies tutorial of the website *https://kaldi-asr.org* with the tools of *Sketch Engine* [8]. Since the result contained some unwanted artefacts like non-linguistic symbols and digits that needed to be either removed or spelled out, additional preprocessing is applied to create a corpus in an appropriate format. For that, we

have written a Python script that detects undesired symbols and substitutes or deletes these.
Then, language models of different orders, namely unigram, bigram and trigram models, are created for each of the three corpora. To find an appropriate weighting of the obtained language models, we used the interpolation script *ami_train_lms.sh* (found in the directory *kaldi/egs/ami/s5b/local*).
Furthermore, the smoothing techniques kneser-ney and witten bell discounting are compared.

3.2. Pronunciation Dictionary

The pronunciation dictionary is also an aspect we have worked on in order to improve the performance of the ASR model. To achieve this goal, we have employed the pronunciation dictionary from the baseline model and have built on it by adding new entries and making modifications.
The baseline pronunciation dictionary has exactly 206510 pronunciations, which is a big number. However, there are still words that occur in the training text but not in the pronunciation dictionary. We have written a Python script to extract the missing items and have later added such 65 items to the pronunciation dictionary. Some of these 65 items are compound words, so we could just give them pronunciations according to those already present in the baseline pronunciation dictionary. For the non-compound words, we annotate them according to the annotation rules from the baseline dictionary.
Here, we note that in the baseline dictionary, every word has corresponding lexical stress, the 0 following the phoneme means no stress, while 1 and 2 means primary stress and secondary stress. For comparison purposes, we have also generated a pronunciation dictionary without lexical stress information. The performance difference is shown in the results section of the report.
Apart from adding missing items, we also tried to modify the pronunciation dictionary according to the accent of the speakers. For example, Russian speakers tend to neutralize the distinction between pat/pet and realize the vowels in the two words both as /ɛ/[9]. To make an adaptation to the training data, what we do is replacing each occurrence of /æ/ with /ɛ/ and add the new pronunciations to the dictionary. There are two ways to do this, one way is to replace the relevant phoneme, /æ/ in this case, in the whole pronunciation dictionary. This is not advisable because it will make the already big pronunciation dictionary considerably larger and slow down the training and testing time. Hence, we simply apply the replacement process on the words that actually occur in the training data. This will make sure the new pronunciation dictionary does not have a too big change in size. There are also other modifications we have done to the pronunciation dictionary as will be discussed in the results section.

3.3. Acoustic Model

In order to compare the performance of a speaker specific acoustic model on only little data and a speaker independent, but more general acoustic model, a model trained on the TED-LIUM corpus is built. As already described in Section 3.1 TED-LIUM consists of scientific talks and therefore fits the domain the acoustic model is intended for. Furthermore, the talks are given in English by people with different native languages and accents. Hence, the challenge of dealing with non-native speech is addressed.

4. Results

4.1. Language Model

Table 1 shows the performance between language models built on the TED-LIUM corpus, the Pavel's kaldi tutorial transcripts and the small corpus consisting of the *kaldi for dummies tutorial* on the kaldi website. As can be seen here, the best performance for the single corpus models is achieved on TED-LIUM (best: 51.55%, trigram model), followed very closely by including a language model on the training data of Pavel's kaldi tutorial transcript (best: 55.27%, bigram model). The weakest performance exhibits the language model on the small corpus of the kaldi for dummies website tutorial (best: 65.43%, trigram model).
The performance of the best single corpus language model on TED-LIUM is outperformed by interpolating the created models (best: 48.7%, trigram model). For that, the optimal lambdas (i.e. the weights) are shown in Table 2. The highest scores by only adapting the language model are obtained by using a trigram model with interpolated Kneser-Ney smoothing. Here, the best triphone model achieves a WER of 48.7%. The best monophone model shows a WER of 64.62% for all models regardless of the order of the language model.

Table 1: *Comparison in % of the best triphone model performances trained with unigram, bigram and trigram language models on different corpora.*

	Unigram	Bigram	Trigram
Pavel's kaldi tutorial	60.53	55.27	55.39
TED-LIUM	61.64	53.59	51.55
Kaldi Tutorial Website	69.21	65.30	65.43
Interpolated Models	57.13	49.19	**48.7**

Table 2: *Optimal weighting for interpolating the trigram language models built on the training data of Pavel's kaldi tutorial, TED-LIUM and the kaldi tutorial of the website.*

	Optimal lambdas
Pavel's kaldi tutorial	0.47977
TED-LIUM	0.511499
Kaldi Tutorial Website	0.00873116

The performance does not differ between the use of interpolated kneser-ney and the witten bell disounting smoothing.

4.2. Pronunciation Dictionary

We have tested the performance of multiple dictionaries, they are all compared with the performance of baseline dictionary.

Table 3: *WER in % for the triphone and monophone model before and after adding missing items.*

	Monophone model	Triphone model
Baseline	64.62	64.93
Baseline Stress removed	64.62	63.32
Missing items added	64.62	59.98
Missing items added Stress removed	64.62	62.02

Table 3 gives us a general idea of the performance of different dictionaries. Apparently by adding missing items, we have a minor improvement of performance. Nevertheless, it's not that clear whether the dictionary with stress or without would have a better performance. Before adding missing items, the result shows that removing stress would have a slightly better performance. However, with the addition of missing items, it seems that retaining the stress information gives us a better model.

Table 4: *WER in % for the triphone and monophone model after adaptation to speaker accent.*

	Monophone model	Triphone model
Missing items added æ/ɛ adapted	64.62	59.73
Missing items added æ/ɛ adapted stress removed	64.62	59.60
Missing items added u:/ʊ adapted	64.62	60.29
Missing items added u:/ʊ adapted Stress removed	64.62	61.21
Missing items added v/w adapted	64.62	59.42
Missing items added v/w adapted Stress removed	64.62	60.16

Table 4 gives some experimental results about the adaptation of the pronunciation dictionary. We have already explained the æ/ɛ adaptation in previous section, so we will omit this point here. However, we want to point out that in this case, the dictionary without stress would have a better performance.
One adaptation we have done is to replace /u:/ with /ʊ/ as Russian speakers tend to neutralize the distinction between the vowel in word pairs such as look/loop[6]. Note that Russian speakers do not actually realize the relevant vowel exactly as the English /ʊ/, but for our convenience, we will just use /ʊ/ in the pronunciation dictionary to avoid further complication. The result might not be what we want because while doing this adaptation the performance will improve when there is no stress (62.02%/ 61.21%), the performance decreases when there is stress (59.98%/ 61.21%). This means we might not want to have this adaptation in the final combined model.
Another adaptation applied is replacing the /v/ with /w/ since we observed that the speaker from the training data tends to pronounce /v/ as /w/ in word like environment. The result shows our instinct is correct because the performance indeed improves, even though the change is quite minimal.
Considering only pronunciation dictionary, the best WER we obtain if we use a dictionary without stress is 59.29% (all three adaptations are present in this case). The best WER we could achieve using a dictionary with stress (æ/ɛ adaptation, v/w adaption both presented) is 59.11%.
To conclude this part, we want to add that through multiple testing, we discovered that by retaining stress, adding missing items along with the æ/ɛ adaptation, v/w adaption, we will obtain the integrated model(including the improved language model) with the best performance, u:/ʊ adaption is discarded because the integrated model with this adaptation does not have the best performance.

4.3. Acoustic Model

The acoustic model trained on TED-LIUM data achieves a performance of 75.03% for the triphone model and 67.1% for the monophone model. This is for the triphone model a performance drop of about 10% compared to the speaker specific baseline.

4.4. Models combined

To generate the integrated model, we simply combine the best pronunciation dictionary with the best language model. The more general acoustic model on TED-LIUM performs worse than the baseline, so we have not integrated it into the final model. The best WER that could be achieved for decoding on the development set is 41.33%. The performance of this model on the test set is with 36.78% a bit better.

5. Discussion

Three main challenges are tackled in this project: data sparsity, domain specific and accented speech. This is done by modifying the language model, pronunciation dictionary and acoustic model.
Language Model Interestingly, the biggest improvement is achieved by tweaking the language model. One possible reason for that could be that a rather weak performance of the acoustic model is, at least for some part, compensated by an appropriate language modeling. This follows from the assumption that a good language model assigns nonsensical word combinations low probabilities and therefore limits the prediction to more reasonable choices. Of course this is only working if the other ASR components deliver results that are meaningful enough for the language model to build on. Noteworthy is also that the smoothing technique does not seem to make a difference. This could arise of the fact that a good language model covers all the relevant words and phrases and therefore, smoothing plays, if any, only a very subordinate role. Making sense is also that the unigram models perform significantly worse than bigram or trigram models and of course the interpolated ones since context is completely disregarded in the case of unigrams.
Acoustic Model In contrast to the improvements achieved by integrating a suitable language model, replacing the sparse acoustic baseline model with a more general one does not lead to better results, but to a decrease in performance of about 10%. Assuming that each speaker has a unique speaking style and accent (because of being a non-native speaker, having a regional accent or similar), it makes sense that integrating a more general model does worsen the performance. While a general domain-specific language model might be statistically more meaningful, the acoustic model is possibly not. This is due to the fact that blending different accents and speaking styles does most likely not lead to more representative acoustic features.
Pronunciation Dictionary In comparison with the performance improvement after tweaking the language model, the effort on pronunciation dictionary is not as fruitful. A performance increase of around 6% (64.62%/ 59.11%) is the best we have been able to achieve. Also note that by tweaking a single phoneme, the performance increase is minimal if not negligible at all. One possible path to further improve the pronunciation dictionary is to make more modifications on the pronunciation dictionary to suit the speaker even further. As for the accent of the speaker, we have seen that the speaker shows indeed some typical Russian accent features. However, we are not sure why u:/ʊ adaptation has an inconsistent result as it worsens

the model when stress is present and behaves contrarily when stress is non-existent. This also leads to the curious question of whether to employ a pronunciation dictionary with stress in the ASR system. The evidence is inconclusive in our case as the best model with stress and the best model without stress do not show much difference in performance (59.11%/ 59.29%).

6. Conclusion

A suitable language model constitutes an important part of a traditional ASR system. In this project, interpolating three language models on domain-relevant data on mainly spoken speech, namely the TED-LIUM transcripts, the training data of Pavel's kaldi tutorial and the corpus of the kaldi for dummies website tutorial, alone improves the WER of about 16% on the development set. In contrast to that, replacing the speaker-specific acoustic model built on sparse data with a more general one does not lead to further improvement, even though the TED-LIUM acoustic model is considered suitable because of containing non-native scientific speech. On the contrary, the WER decreases by around 10%. As for pronunciation dictionary, adding missing items and adapting towards the speaker specific accent leads to a performance improvement of around 6%. After applying the described modifications, the best WER obtained when using our model to decode on the test set is 36.78%.
We hope this is a sufficient result and we would like to build on the knowledge and experience we accumulated through this project in our future work and studies.

7. References

[1] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Vesel, "The kaldi speech recognition toolkit," *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 2011.

[2] S. Broman and M. Kurimo, "Methods for combining language models in speech recognition." 2005, pp. 1317–1320.

[3] S. Young, "Large vocabulary continuous speech recognition: A review," *IEEE Signal Processing Magazine*, vol. 13, no. 5, pp. 45–57, 1996.

[4] G. Hunter, "Statistical language modelling of dialogue material in the british national corpus." Ph.D. dissertation, 2004.

[5] I. Thompson, "Foreign accents revisited: The english pronunciation of russian immigrants," *Language learning*, vol. 41, no. 2, pp. 177–204, 1991.

[6] O. Bondarenko, "Does russian english exist," *American Journal of Educational Research*, vol. 2, no. 9, pp. 832–839, 2014.

[7] A. Rousseau, P. Deléglise, and Y. Estève, "TED-LIUM: an automatic speech recognition dedicated corpus," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA), may 2012, pp. 125–129. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2012/pdf/698_Paper.pdf

[8] A. Kilgarriff, P. Rychlý, P. Smrž, and D. Tugwell, "Itri-04-08 the sketch engine," *Information Technology*, 2004.

[9] "Non-native pronunciations of english," Aug 2021. [Online]. Available: https://en.wikipedia.org/wiki/Non-native_pronunciations_of_English#Russian