Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200623

OCR Challenges for a Latvian Pronunciation Dictionary

Laine STRANKALE, Pēteris PAIKENS
University of Latvia, Institute of Mathematics and Computer Science

Abstract. This paper covers the devlopment of a custom OCR solution based on the Tesseract open source engine developed for digitization of a Latvian pronunciation dictionary where the pronunciation data is described using a large variety of diacritic markings not supported by standard OCR solutions. We describe our efforts in training a model for these symbols without the additional support of preexisting dictionaries and illustrate how word error rate (WER) and character error rate (CER) are affected by changes in the dataset content and size. We also provide an error analysis and postulate possible causes for common pitfalls. The resulting model achieved a CER of 2.07%, making it suitable for digitization of the whole dictionary in combination with heuristic post-processing and proofreading, resulting in a useful resource for further development of speech technology for Latvian.

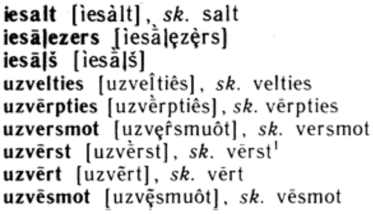
Keywords. OCR, pronunciation, Tesseract

1. Problem Description

Accurate speech recognition and speech synthesis are greatly helped by a large database of words and their phonetic notations. For Latvian the most comprehensive resource of phonetic information currently available is *"Latviešu valodas pareizrakstības un pareizrunas vārdnīca"* (LVPPV)[1], "Latvian spelling and pronunciation dictionary", which contains over 80000 words with full pronunciation transcription. Unfortunately, a machine-readable version of this dictionary is not available because of historical reasons of how this dictionary was originally developed.

During earlier digitization efforts, the LVPPV was scanned. As seen in Figure 1, each dictionary entry contains a pronunciation section enclosed in square brackets. However, in the earlier digitization only the spelling part of the entries was suitable for OCR technologies available at that time, as the pronunciation information is encoded in a custom set of symbols not expected by the available OCR models.

The availability of this scanned data and the requirement for a large machine-readable resource of Latvian pronunciation motivates our research to develop an accurate OCR model for this custom phonetic alphabet to assist a full digitization of this dictionary. Although OCR results are rarely perfect and would be expected to contain mistakes that need human review, an effective custom OCR model

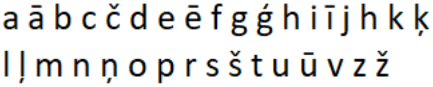


iesalt [ìesàlt], *sk.* salt
iesāļezers [ìesà̄ļȩzȩ̀rs]
iesāļš [ìesà̄ļš]
uzvelties [uzvel̂tiês], *sk.* velties
uzvērpties [uzvḕrptiês], *sk.* vērpties
uzversmot [uzvȩr̂smuôt], *sk.* versmot
uzvērst [uzvḕrst], *sk.* vērst^1
uzvērt [uzvẽrt], *sk.* vērt
uzvēsmot [uzvȩ̃smuôt], *sk.* vēsmot

Figure 1. Extracts from Latvian spelling and pronunciation dictionary (LVPPV)

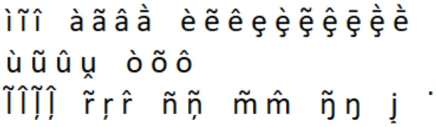
would ensure that this data can be reviewed and corrected with a reasonable amount of manual work.

In LVPPV the pronunciation is denoted by using symbols that extend the standard Latvian alphabet (Figure 2) with additional symbols with diacritic modifiers illustrated in Figure 3. Accurate recognition of this phonetic alphabet is challenging because meaningful differences in pronunciation are expressed by minor variations of diacritic marks – for example, there is a common need to distinguish the pronunciation of the letter ‘a’ as ‘ã’ or ‘â’. Standard OCR systems distinguish visually similar characters through assistance of dictionaries and statistical n-gram frequency models, which is not possible in this scenario as we need to digitize the only dictionary which has this data.



a ā b c č d e ē f g ģ h i ī j h k ķ
l ļ m n ņ o p r s š t u ū v z ž

Figure 2. Symbols in Latvian alphabet (only lowercase shown)



ì ĩ î   à ã â à̄   è ẽ ê ȩ ȩ̀ ȩ̃ ȩ̂ ȩ̄ ȩ̄̀ ḕ
ù ũ û ṷ   ò õ ô
l̃ l̂ ļ̃ ļ̂   r̃ ŗ r̂   ñ ņ̃   m̃ m̂   ŋ̃ ŋ   j̭   ˙

Figure 3. Additional symbols in phonetic transcriptions (LVPPV uses only lowercase letters)

2. Related Work

There has been no known work in digitizing this style of printed phonetic transcription of Latvian. There is phonetic transcription available for a limited number of words in digital resources *“Mūsdienu latviešu valodas vārdnīca” Dictionary of contemporary Latvian* (MLVV) and tezaurs.lv online dictionary [7]. These resources are much smaller than LVPPV, but they can be used as a test set for verifying the OCR accuracy on the subset of words contained in both resources.

There is earlier work on rule-based approaches to derive phonetic transcriptions of Latvian[5]. However, these methods are limited and would also directly benefit from a larger digital database of phonetic transcriptions such as the final result of digitizing LVPPV.

In reviewing research on OCR technology applications, we were not able to identify any useful publications on digitizing phonetic transcription of other languages. However, there is extensive literature on the more general task of developing OCR solutions for new scripts. The prevailing paradigm for such OCR systems relies on retraining existing general-purpose OCR systems on a targeted set of training data examples for the new script using supervised machine learning and deep neural networks. While there are also examples using custom systems implemented from scratch, especially for commercial solutions, we consider that it is reasonable to adapt an existing system in order to reuse existing functionality of recognizing the latin alphabet and only adjust the specific characters (diacritic combinations) that are used in this pronunciation dictionary.

The two leading OCR tools that support training additional languages are Abby Finereader[1] and Tesseract[6]. For this research we have chosen Tesseract as it is a free open-source solution and has been the basis for multiple successful implementations of OCR for a new script[2,4].

3. Method

Since version 4 Tesseract uses a LSTM[3] based neural network architecture which significantly outperforms previous versions. Tesseract 4 provides three training methods[8]:

1. fine-tuning for impact (adding a few extra characters to an existing model);
2. training just a few layers (removing the top layers from an existing model and replacing with new ones);
3. training from scratch.

These approaches differ substantially by how much training data is required. As the pronunciation data introduces a significant number of new characters and we want to limit training data size, the second training method is considered the most appropriate. LVPPV largely contains characters and letter patterns characteristic to the Latvian language, thus Tesseract's pre-trained Latvian language model was chosen as the base model.

3.1. Data Preparation

The Tesseract training process requires training data that consists of scanned images annotated with character bounding boxes aligned with appropriate characters. In this work, whole pages of the scanned LVPPV dictionary were used as the basic units of data. The input data was selected, so that the chosen images contain all the new phonetic characters frequently enough to minimize errors in the final model.

[1] https://www.abbyy.com/en-us/finereader/

OCR requires good quality images and LVPPV was scanned in 600 dpi resolution. Additionally, as the goal of this research is to recognize text from this particular dictionary, we know that all data will be a single font, thus we did need to adapt it for font variation as is commonly required for general purpose OCR solutions.

3.2. Post-processing Heuristics

We developed a method to determine a particular character's error rate. It uses Levenshtein distance, also known as edit distance, that measures the difference between two sequences, in this case, the expected character string and the OCR generated character string.

This allowed to both strategically choose pages from LVPPV for training data and detect instances where rule-based post-processing could be applied using known phonetic transcription characteristics and gained insights.

Analysis of OCR errors during development revealed that a common error pattern involves character duplicates – for example, 'ã' which was mistaken for 'ãâ'. Some examination revealed that this is an unresolved issue in Tesseract implementation wherein if two characters have a similar probability of being the correct then both are output[2]. Some proposed solutions suggested looking at the character bounding boxes but because of the way Tesseract is implemented this still would not be a complete solution. However, as the model improves, the frequency of these errors should decrease. A large part of such errors can be automatically corrected with heuristic post-processing methods because these mistakes generate character sequences that are not plausible in Latvian words.

4. Training

We performed experiments to analyze the effect of training set size on OCR accuracy to determine when the effort put into preparing training examples exceeds the effort needed to check for errors in the processed pages manually.

Multiple models were trained with various limitations on training data size. Data was increased gradually, with one LVPPV page as the step size. The final model was trained on a dataset of 2949 lines (dictionary entries with a word spelling, its pronunciation, and auxiliary comments); Table 1 illustrates how the LVPPV page count corresponds to the line count in the training data.

Table 1. Training Set Size. Line Count vs. LVPPV Page Count

Line Count	1175	1314	1433	1551	1666	1784	1910	2026
LVPPV Page Count	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*
Line Count	2141	2256	2372	2483	2601	2716	2832	2949
LVPPV Page Count	*18*	*19*	*20*	*21*	*22*	*23*	*24*	*25*

The dataset was split into three parts; separate training and test sets were used during development to train the models and choose when to stop the training

[2] https://github.com/tesseract-ocr/tesseract/issues/2738

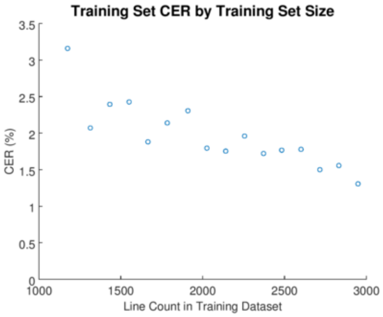


Figure 4. Training Set Character Error Rate

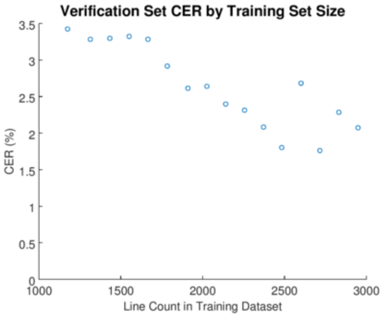


Figure 5. Verification Set Character Error Rate

process, while a separate verification set was used in final accuracy assessments for this paper.

It is important to note that we must evaluate not only the OCR accuracy of the pronunciation OCR. LVPPV entries contain both regular Latvian words and phonetic transcriptions and both categories need to have a high accuracy - it is not sufficient to obtain a correct pronunciation if it can not be automatically mapped to the proper dictionary word because of an OCR mistake in the spelling part.

5. Result Assessments

It was expected that (a) larger training size would correspond to a smaller error rate, and (b) increased character frequency would decrease the character's error rate.

Initially we trained some models for exploratory research to eliminate and fix any persistent errors in character set and input data. For the experimental data included in this paper, 16 models were trained with different amounts of training data, from 10 pages (1175 lines) to with 25 pages (2949 lines).

The analysis uses *character error rate* CER and *word error rate* WER to measure the quality of the trained models. Figures 5, 7 show that both rates fell with the increase in the training set. However, the trend is not consistent and has some significant outliers. Overall, error rate patterns in training and verification sets are similar, although, as expected, the training set has higher accuracy.

5.1. Character Level Analysis

On average, the total error rates fell which support expectation (a), however, character level errors exhibit a different pattern. An important hypothesis was the question of whether a particular character's error rate would decrease with the increase of its frequency in the training set.

Table 2 shows the character frequencies in the training data in the various experiments as more pages of training data were added. Additionally, by examining the verification set character errors (insertions, deletions, and substitutions) in the table 3 a few things can be noted:

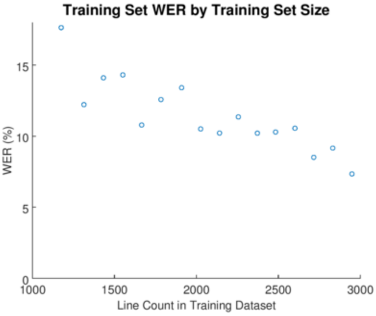


Figure 6. Training Set Word Error Rate

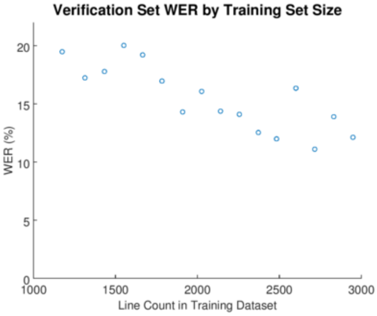


Figure 7. Verification Set Word Error Rate

Table 2. Character frequencies in the training set after the addition of n-th page to the dataset

N-th Page Added	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**
Symbol										
ì	228	231	250	252	262	269	270	270	373	407
î	132	134	142	147	152	152	154	164	183	193
ĩ	195	218	236	244	260	268	279	285	292	319
ê	269	277	296	303	305	311	314	346	353	358
ẽ	210	227	260	268	271	276	284	296	310	326
ȩ	105	105	106	107	108	108	109	127	128	128
è	102	104	104	113	115	118	119	124	125	125
ȩ̃	100	100	101	101	101	101	101	112	112	124
l̂	21	21	22	22	33	33	40	50	54	59
l̃	33	39	40	40	54	55	105	124	127	172
ã	180	196	202	227	234	260	284	288	288	296
â	100	101	103	104	106	109	110	116	124	125
a̍̀	8	8	8	12	12	13	13	13	23	23
à	71	73	74	103	104	106	106	109	110	116

(a) The general trend is towards fewer errors as training data increases, particularly for characters with a smaller diversity of diacritic marks like ”r”

(b) Larger frequency does not consistently correspond to fewer errors. For instance, the frequency of *letter l with tilde* almost doubled when page 22 was added, however, its error actually increased and the letter was consistently not being recognized.

(c) An inappropriately large frequency can cause ’overconfidence’. This can be seen in the last models for the letter ”ì” where it went from being missed to being overused in place of ”i”

As exemplified, the data illustrates an instability of the models towards different specific errors, as training a new model on a different, larger set of training data may result in a very different pattern of errors, possibly because of random initialization effects on the neural network model.

In this particular application, a significant subset of errors could be easily corrected in heuristic post-processing as some OCR output character combinations are not used in Latvian pronunciation. It might also be plausible to use a small statistical language model for this disambiguation, but due to technical difficulties (limited quantity of available data, and this would need to be a subword model for short character n-grams inside a single word) of integrating this modification, it was not attempted at this time.

Table 3. Most common character errors in verification set by training set size

Pages		**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**
OCR	**GT**												
è	ê	2	3	8	2	2	1	2	2	3	4	6	3
ẽ	ȩ̃	19	0	0	0	2	2	6	0	6	0	13	1
e	ȩ	9	5	2	3	5	6	3	1	5	4	11	6
ẽ	ê	4	9	4	15	6	6	7	3	27	7	4	1
ã	à	9	11	10	10	13	12	8	13	11	11	11	8
â	à	48	5	3	7	1	1	5	0	17	4	21	2
l̃	l	2	0	0	0	1	1	5	0	2	1	8	20
l̃	l̂	1	0	4	0	14	0	3	12	0	3	2	14
l	l̂	18	1	3	20	7	9	2	4	2	2	0	2
l̂	l̃	0	57	10	0	0	21	5	1	40	2	6	0
l	l̃	25	6	19	36	18	21	14	5	13	17	2	4
î	ì	15	4	4	12	10	1	3	5	1	1	1	1
ì	i	12	16	34	10	19	0	0	18	0	0	10	31
i	ì	29	28	19	21	22	34	46	27	37	21	19	13
CER		*3.28*	*2.92*	*2.61*	*2.64*	*2.40*	*2.31*	*2.08*	*1.80*	*2.68*	*1.76*	*2.29*	*2.07*

6. Conclusion

Our research demonstrates that freely available OCR models can be adapted for a new, specific set of diacritic markings with reasonable accuracy (2.07% character error rate) even with no availability of preexisting dictionaries or language models and only a small quantity of training data, 25 pages in our experiments.

However, our error analysis indicates that the training of Tesseract OCR models is not clear cut: more data might not necessarily mean better accuracy for any specific erroneous characters. Care should be given to the frequencies of characters in the training data to avoid detrimental effects, and a specific character's accuracy can fluctuate widely.

Whether these effects can be diminished by using other training methods could be explored by testing different OCR engines and their versions or experimenting with character frequencies in the training data.

The resulting OCR models will be applied to the full LVPPV data and, as the manual post-processing and proofreading work is finalized, result in up to 80000 phonetic transcriptions added to the publicly available lexical resources for Latvian. This data will facilitate the development of accurate speech synthesis and speech-to-text solutions for Latvian.

Acknowledgements

This research work was supported by the Latvian Council of Science, project “Latvian WordNet and word sense disambiguation”, project No. LZP-2019/1-0464.

References

[1] Porīte T. Raģe S. Ceplītis L., Miķelsone A. *Latviešu valodas pareizrakstības un pareizrunas vārdnīca.* ”Avots”, 1995.

[2] Md Hasnat, Muttakinur Rahman Chowdhury, Mumit Khan, et al. Integrating bangla script recognition support in tesseract ocr. 2009.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[4] Isabell Hubert, Antti Arppe, Jordan Lachler, and Eddie Antonio Santos. Training & quality assessment of an optical character recognition model for northern haida. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, pages 3227–3234, 2016.

[5] M. Pinnis and I. Auzina. Latvian text-to-speech synthesizer. In *Human Language Technologies - The Baltic Perspective*, volume 219. IOS Press, 2010.

[6] Ray Smith, Daria Antonova, and Dar-Shyang Lee. Adapting the tesseract open source ocr engine for multilingual ocr. In *Proceedings of the International Workshop on Multilingual OCR*, pages 1–8, 2009.

[7] A. Spektors, I. Auzina, R. Dargis, N. Gruzitis, P. Paikens, L. Pretkalnina, L. Rituma, and B. Saulite. Tezaurs.lv: the largest open lexical database for latvian. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*, 2016.

[8] Tesseract. How to use the tools provided to train tesseract 4.00. https://tesseract-ocr.github.io/tessdoc/TrainingTesseract-4.00. Accessed: 2020-04-03.