S. Krajči (ed.): ITAT 2018 Proceedings, pp. 35–40
CEUR Workshop Proceedings Vol. 2203, ISSN 1613-0073, © 2018 Oldřich Krůza

Phonetic Transcription by Untrained Annotators

Oldřich Krůza

Charles University, Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics
kruza@ufal.mff.cuni.cz

Abstract: The paper presents an application for lay, untrained users to generate high-quality, aligned phonetic transcription of speech. The application has been in use for several years and has served to transcribe over 600 thousand word forms over two versions of a web interface. We present measures for compensating the lack of expert training.

1 Introduction

1.1 Our Setting

The work presented in this paper is a part of the project that tends the spoken corpus of Karel Makoň[1]. The corpus is of the single speaker and has been recorded in amateur conditions, while the author was speaking to his friends about a novel way to interpret the teaching of Jesus and of mystic and spirituality in general. Karel Makoň died in 1993 and a community of favorers of his teachings has persevered since then.

The talks can be seen as companions to Makoň's written works. Together they form a unique, extensive, consistent systematization of the spiritual path tailored to modern westerners and accessible primarily to Czech speakers. It draws heavily on traditional Christian mysticism as well as ancient tradition of India and China, adapting them for the present. The whole system can be seen as a manual for entering the eternal life prior to the physical death.

There are over 1000 hours of digitized recordings of Karel Makoň, they are accessible under the CC-BY license and the project aims at bringing the most benefit out of them. The first step was digitizing the recordings from the original magnetic tapes, the second step was releasing all of them on the world-wide web, the third step was developing a web-based system for human / machine transcription of the bulk, allowing for search.

The transcription we do is both phonetic and orthographic.[1] Our users are supposed provide orthographic transcription where the pronunciation is standard and phonetic otherwise.

[1]There is no actual focus on orthography. Instead, we mean the natural way of transcribing the speech to human-readable text. Where it matters, focus is directed at precise correspondence with the utterances instead of language cleanliness.

1.2 Architecture Overview

The system consists of

1. The corpus in compressed audio format. We use mp3 and ogg/vorbis to accomodate most browsers. These data are hosted on an external CDN.
2. The exact copy of the corpus in parametrized (MFCC) format. These data reside on the back-end server.
3. A complete, aligned transcription of the recordings, hosted on the back-end server and mirrored on a CDN.
4. Acoustic model trained on the human-transcribed part of the corpus.
5. Language model trained using Srilm[2] on a combination of publicly available Czech texts, Karel Makoň's written works, and both the human-submitted and automatically-acquired transcription.
6. Back-end API for collecting correcions to the transcription, serving the transcription and allowing full-text search with elasticsearch[2].
7. Separately hosted front-end web application serving as an interface for playing the recordings, synchronously displaying the transcriptions and collecting the corrections from users.

To get the initial transcription, we have manually transcribed some 10 minutes of the material using Transcriber[3], trained an acoustic model on it and recognized the whole data using it.

2 Annotator Expertise

Our case is on the edge of what can be called linguistic data annotation. In our lucky part of the world where alphabetization nears 100%, transcription of speech is hardly expert work. On the other hand, ensuring that the transcription exactly matches the audio

[2]https://www.elastic.co/products/elasticsearch
[3]http://trans.sourceforge.net/

- as a representation of the words uttered and of their meaning,
- on the phonetic level, phone for phoneme,[4]
- on the time axis

is beyond what can be expected from an untrained user.

Linguistic data annotation in general requires trained personnel. If we only look at the Prague Dependency Treebank, we can notice the annotators provided such a degree of expertise they have become the co-authors[3].

Crowdsourcing, community-driven approach or engaging volunteers is an ever stronger, popular way of obtaining assets that would otherwise be unbearably costly. Let us mention for example Mihalcea (2004)[4] who delegates word-sense disambiguation to volunteers. The Wikicorpus[5] as well as the MASC[6] gather annotation from volunteers.

In most cases, quality is very important for data annotation, so some kind of control is essential, no matter how expert the annotators. Trivially, the less expertise, the more control is needed.

2.1 Quality Control

A common way of dealing with quality control is to inspect annotator agreement. This has the huge downside that every piece of data must be annotated at least twice, which reduces the yield by 50+%.

There is another reason not to use it in our case. Our application is designed for people who want to listen to the recordings out of interest and their contribution to the quality of the transcription is more of a by-product. It would be hard to convince them to choose exactly a recording that another user has already transcribed.

Luckily, we can implement automatic measures to aid the annotators to deliver higher-quality transcription.

2.2 Forced Alignment

We always assume an existing transcription, so we can see the user's contribution as a correction. Every submission has the form of replacing a text segment with another. Since the transcriptions are time-aligned to the audio, we also know exactly what is the corresponding audio segment to the text submitted.

This enables us to perform forced alignment on the submitted text and the audio. With a well selected pruning threshold, we can distinguish false transcriptions and reject them, providing feedback to the contibutor. Since every segment of audio fits the acoustic model to a different degree, both false positives and false negatives will inevitably occur.

False positives (when the system accepts a wrong transcription) present a problem, since the error will enter the training data set. But users can often circumvent false negatives by submitting the transcription divided in different segments. Of course, this method can also be used to force a wrong transcription but we assume no malevolence on the part of the users.

Apart from catching wrong transcription, the forced alignment mechanism provides exact synchronization on the time axis. This is a completely missing element in the case of virtually all programs for computer-aided transcription. For some examples, Transcriber, a veteran open-source transcribing program for Linux, expects the user to provide alignment on the level of phrases; Transcribe,[5] a commercial web-based transcribing tool, allows the user to add timestamps anywhere in the text. There is no acoustic model, hence nothing to match against.

3 Phonetic Transcription

3.1 Purpose

We have originally built the acoustic model using HTK,[6] the Hidden Markov Model Toolkit. Here, explicit phonetically labeled training data are necessary for training. We are switching to DNN, using Mozilla's DeepSpeech,[7] where no explicit phonetic annotation is needed but for some purposes like forced alignment, the original HMM is still irreplaceable.

Also, the phonetic labeling is valuable per se for research purposes.

3.2 Phoneme Set

We use a subset of PACal[7]. We shall also refer to individual phonemes in this paper using the PACal notation in `monospace font`. For reference, Table 1 lists the phonemes used with their IPA notation.

3.3 Acquisition

The phonetic transcription is in normal case also a product of forced alignment, as in case of pronunciation variants, it selects the most fitting one. This requires a way to automatically obtain all pronunciation variants of any word. We use a combination of a rule-based system inspired by Psutka et al.[8], in combination with a dynamic dictionary. The dynamic dictionary is a list of alternative pronunciations of a word, which expands as the app is being used.

[4] In the sense that each written phoneme corresponds to exactly one uttered phone.

[5] https://transcribe.wreally.com/

[6] http://htk.eng.cam.ac.uk/

[7] https://github.com/mozilla/DeepSpeech

IPA	PACal	common grapheme	IPA	PACal	common grapheme
a	a	a	ɱ	mg	tra͟mvaj
aː	aa	á	n	n	n͟e
aʊ̯	aw	au	ŋ	ng	tan͟k
b	b	b	ɲ	nj	ň
t͡s	c	c	o	o	o
t͡ʃ	ch	č	oː	oo	ó
d	d	d	oʊ̯	ow	ou
ɟ	dj	ď	p	p	p
d͡z	dz	dz	r	r	r
d͡ʒ	dzh	dž	r̥̊	rsh	t̥r̥i
ɛ	e	e	r̥	rzh	ř̥íz
ɛː	ee	é	s	s	s
eʊ̯	ew	eu	ʃ	sh	š
f	f	f	t	t	t
g	g	g	c	tj	ť
ɦ	h	h	ʊ	u	u
i	i	i	uː	uu	ú, ů
iː	ii	í	v	v	v
j	j	j	x	x	ch
k	k	k	z	z	z
l	l	l	ʒ	zh	ž
m	m	m͟ák		sil	
				sp	

Table 1: Phonemes used in transcription

The users are instructed to transcribe any words with non-standard pronunciation phonetically and then correct their orthographical form. This is one of the few cases where we are coercing the users to something.

When the orthographically broken, phonetic transcription of a word is submitted, if it passes the forced-alignment phase, it is integrated into the displayed transcription. The word's data representation consists of its

1. occurrence: the word as it appears in the text, including capitalization and punctuation,
2. wordform: the word as it appears in the language model and phonetic dictionary (computed as the occurrence in lowercase and stripped of non-alphabetic characters[8]),
3. pronunciation: an array of phonemes,
4. timestamp: distance of the beginning of the word from the beginning of the file, in seconds, in precision of 2 decimal digits,
5. manual/automatic: boolean flag denoting whether the word has been transcribed manually or not,
6. confidence measure: in case of automatically acquired words, the confidence-measure score of the recognizer.

Once merged into the displayed transcription, each word's occurrence can be edited manually. Now the user can enter the correct form deviating from Czech pronunciation rules.

Doing so results in adding the wordform-pronunciation couple to the dynamic pronunciation dictionary and is also used for forced alignment. Thus, this operation need only be performed once per word and any subsequent time the word is entered in its standard orthographic form, the correct pronunciation is inferred.

For example, let's examine the scenario of transcribing the sentence *Proč se toto nestalo Marii Markétě Alacoque? (Why hasn't this happened to Mary Margaret Alacoque?)* Its phonetic representation is `p r o ch sp s e sp t o t o sp n e s t a l o sp m a r i j i sp m a r k ee tj e sp a l a k o k sil .`

1. Suppose the user enters the correct ortographic transcription.
2. The phonetic transducer outputs `p r o ch sp s e sp t o t o sp n e s t a l o sp m a r i j i sp m a r k ee tj e sp a l a c o k v u e sil .`

[8] This implies that all non-alphabetic characters are always a part of a token and never form a token on their own.

3. With a bit of luck, the forced alignment fails because of the distiction of the phone sequence `k o k` and `c o k v u e`.
4. The transcription is rejected, the user realizes that the word is pronounced in a non-standard way and re-tries with *Proč se toto nestalo Marii Markétě alakok?*
5. Forced alignment succeeds now and the entered transcription is merged into the view.
6. The user selects the non-existent word *alakok?* and edits its occurrence to *Alacoque?*
7. Now the word is correctly stored and on any subsequent user inputs of *Alacoque* with any punctuation or capitalization, the pronunciation `a l a k o k` is inferred by the forced alignment.

3.4 Phonetic Respelling

With all advantages of using PACal as a representation for phonemes, it is clearly not the most natural way for lay Czechs to write down and read literal pronunciation. Thanks to the simple, mostly deterministic mapping between phonemes and graphemes, pronunciation respelling is a reliable, natural way. There's not even a need for explicit syllable separation as seen in English pronunciation respelling (wikipedia[9] gives the example *"Diarrhoea" is pronounced DYE-uh-REE-a*). We postulate that the phonetic respelling is natural to all alphabetized native Czech speakers as a fact without any supporting research, based on experience alone.

The previous subsection gave an example of using pronunciation respelling in Czech with the example of *alakok* for *Alacoque*. The direction from the phonetic respelling to the phoneme array is covered by the ortographic-to-phonetic transducer. But we also need the opposite direction to provide the users a way to check whether the pronunciation selected by the forced alignment fits.

For this purpose, we have created a JavaScript module for transduction between the array of phonemes and the pronunciation respelling.[10]

The algorithm is simple. In most cases, a phoneme corresponds uniquely to one character in the respelling. Exceptions are as follows:

1. The phoneme `x` is spelled *ch*.
2. The phonemes `dz dzh` are spelled *dz dž*.
3. The diphtongs `aw ew ow` are spelled *au eu ou*.
4. Sequences `c h, o u, a u, e u, d z, d zh` are spelled *c'h, o'u, a'u, e'u, d'z, d'ž*. Note though, that the sequence `c h` is purely hypothetical, as it contradicts voiced/voiceless assimilation.
5. Voiceless alveolar fricative trill is explicated as *r'*.
6. Palatal nasal and labiodental nasal are spelled *n', m'*.
7. Trailing silence is not represented.

The module includes two-way transduction, although only the one from array of phonemes to human-readable phonetic respelling is needed in our application. Still, the user can mark up special-case pronunciation with the apostrophe, like the sequence of phonemes `o` and `u` with the string `o'u`. The need has never occurred during the six years' lifespan of the application.

Note that when encoding into the phonetic respelling, none of *di ti ni dě tě ně* is ever output. The palatal consonants are always explicitly spelled out and e.g. the sequence `n i` is always spelled *ny*

A few examples of words, pronunciation and phonetic respelling as output by the algorithm (given the corresponding pronunciation is input as phoneme list):

- nic /`nj i c`/: *ňic*,
- kdo /`g d o`/: *gdo*,
- disk /`d i s k`/: *dysk*,
- dřít /`d rzh ii t`/: *dřít*,
- třít /`t rsh ii t`/: *tř'ít*,
- auto /`aw t o`/: *auto*,
- nauka /`n a u k a`/: *na'uka*,
- džbán /`dzh b aa n`/: *džbán*,
- odžít /`o d dz ii t`/: *od'žít*,
- odznak /`o dz n a k`/: *odznak*,
- podzemí /`p o d z e m ii`/: *pod'zemí*,
- noc /`n o c`/: *noc*,
- tento /`t e n t o`/: *tento*,
- hangár /`h a ng g aa r`/: *han'gár*,
- samba /`s a m b a`/: *samba*,
- tonfa /`t o mg f a`/: *tom'fa*.

The use of apostrophe for distinguishing ambiguities and special cases is not 100% intuitive and presents another point where instruction is necessary for the user to use this feature properly.

[9] https://en.wikipedia.org/wiki/Pronunciation_respelling
[10] https://github.com/Sixtease/MakonReact/blob/master/src/lib/Phonet.js

4 Evaluation

We have presented our web application as a tool that enables gathering precisely aligned, phoneme-exact transcription from untrained casual visitors. We have presented measures for reaching this goal but the degree to which it was reached remains unclear.

We have no gold standard data to measure the quality of our manual transcriptions. On the contrary, we use the manual transcriptions as gold standard for the automatic recognition. What we can to, however, is look at some random samples and try to get a rough idea of how the system performs.

4.1 Validation by Forced Alignment

One thing we can examine are the approvals / rejections of the forced alignment. Of 109640 forced alignment attempts, 3419 have failed, which makes for 3.12% rejection rate. We have manually inspected 20 random failed attempts and came to the following numbers:

- 11 cases were false negatives, where the transcription was correct and should have been accepted,
- 4 cases were caused by acoustic irregularities like noise,
- 4 cases were true negatives caused by wrongly chosen segment boundaries and
- 1 case was true negative caused by wrong transcription.

Hence, in 25% of the minimalistic sample, the forced alignment did its job of a validator and prevented a piece of broken training data from entering the dataset. In 55% it was a nuisance and failure, and in the remaining 20%, it rejected a valid transcription but prevented a bad training example from occurring, so we can see this in positive light.

4.2 Non-Standard Pronunciation

We can also track how the scenario described in subsection 3.4 is applied. We have looked up four promising example records in the dynamic dictionary and checked submitted transcriptions containing them. Table 2 lists for each of them the correct orthographic form, the wrong pronunciation obtained by the transducer, the correct pronunciation and finally the phonetic respelling. Each is followed by the number of occurrences in the manually transcribed data.

We can see in Table 3 that the majority of cases results in both orthographic and phonetic forms being correct. Only in about 13% cases, the orthographically incorrect form is kept. We attribute this to the fact that those who use the phonetic respelling are aware of the problematic and mostly go the whole way and clean up.

On the other hand, nearly a third of the cases show the wrong phonetic representation. This is a serious problem on at least two levels: Firstly, it shows that the forced aligner failed to catch the error. Secondly, it lets bad examples into the training dataset.

One of the apparent reasons for this to happen is that the dynamic dictionary only recognizes exact matches. We can see in one file, for example, all occurrences of the form *Weinfurter* to have correct pronunciation while *Weinfurterovi* to have a broken one.

Other factors likely include user carelessness or ignorance, which is exactly what our application is trying to compensate, but fails in these cases.

The cases with false orthographic form don't pose much of a problem. It can harden searching for the term in question but performing a search for the phonetic respelling or even automatically searching the pronunciation would easily mitigate this.

The fourth combination of phonetic respelling and false pronunciation is of course not occuring.

5 Conclusion

We have presented an application that has been providing access to the extensive corpus of Karel Makoň and to acquire an almost complete transcription thereof. Nearly 70 hours corresponding to over 600,000 word forms have been transcribed manually with minimal financial[11] as well as development[12] costs. Only some of the volunteers have indulged instruction time in order of minutes. The rest of the corpus has been transcribed using an ASR system trained on these ever-growing data.

We have presented the ways we use to aid the untrained users to provide a high-quality orthographic and phonetic time-aligned transcription. We have attempted a rough evaluation of the success rate of the measures presented. Though clearly far from perfect, they do serve the purpose and set a baseline for improvements or novel approaches.

The system has been built with the motivation of spreading the message contained in Karel Makoň's talks. However, to make the technology more useful, we are actively looking for similar settings where it could be deployed.

Acknowledgments

The research was supported by SVV project number 260 453.

[11] In early stages, we kept a paid annotator to test the application.

[12] The system has been written by a single developer.

Correct spelling	#	wrong phonetic pronunc.	#	correct pronunciation	#	phon. respel.	#
Moody	2	m o o d i	0	m uu d i	4	*múdy, můdy*	2
Descartes	2	d e s c a r t e s	0	d e k aa r t	4	*dekárt*	2
Weinfurter	30	v e j n f u r t e r	13	v a j n f u r t r	19	*vajnfurtr*	2
Michelangelo	6	m i x e l a ng g e l o	2	m i k e l a n dzh e l o	4	*mikelandželo*	0

Table 2: Examples of non-standard pronunciation in the manually transcribed data

	phonetically correct	phonetically incorrect
orthographically correct	25	15
orthographically incorrect	6	0

Table 3: Success rate for phonetic and orthographic representation of foreign words based on data from table2

This work has been using language resources stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2015071).

References

[1] Jurik Hájek. Český mystik Karel Makoň. *Dingir*, 2007/4:142–143, 2007.

[2] Andreas Stolcke. Srilm-an extensible language modeling toolkit. In *Seventh international conference on spoken language processing*, 2002.

[3] Jan Hajič. Complex corpus annotation: The prague dependency treebank. *Insight into Slovak and Czech Corpus Linguistics. Veda Bratislava*, 2005:54–73, 2005.

[4] Rada Mihalcea and Timothy Chklovski. Building sense tagged corpora with volunteer contributions over the web. *Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003*, 260:357, 2004.

[5] Samuel Reese, Gemma Boleda, Montse Cuadros, and German Rigau. Wikicorpus: A word-sense disambiguated multilingual wikipedia corpus. 2010.

[6] Nancy Ide, Christiane Fellbaum, Collin Baker, and Rebecca Passonneau. The manually annotated subcorpus: A community resource for and by the people. In *Proceedings of the ACL 2010 conference short papers*, pages 68–73. Association for Computational Linguistics, 2010.

[7] Jan Nouza, Josef Psutka, and Jan Uhlír. Phonetic alphabet for speech recognition of czech. *Radioengineering*, 6(4):16–20, 1997.

[8] Josef Psutka, Jan Hajic, and William Byrne. The development of asr for slavic languages in the malach project. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 3, pages iii–749. IEEE, 2004.