ISSN 1870-4069

Restructuring Spoken Corpus for Streaming Emulation

Jan Oldřich Krůza

Charles University,
Institute of Formal and Applied Linguistics,
Faculty of Mathematics and Physics,
Czech Republic

kruza@ufal.mff.cuni.cz

Abstract. This paper shares the experience of transforming the way of storing a spoken corpus into more numerous smaller files. Discussed is the motivation: a change in the usage of the data; the corpus itself: a 1000-hour set of recordings of a single speaker; and decisions to be made for the restructuring, reasons for them and their impact.

Keywords: Speech recognition, cassette digitization, recording on magnetic tapes.

1 Introduction

The way a volume of data is stored impacts all subsequent processing and exploitation [7]. A decision on storing the data has to be made as soon as the data is being acquired. Obviously, the difficulty of changing the way of storing the data increases with the volume of the data stored and with more parties relying on it. The relying parties can be in-house software tools or even public interfaces (I hesitate to say API because we're talking about access to data, not to applications). In my case, the data at hand is a collection of speech recordings of total magnitude of about one thousand hours. The decisions to be made span many levels and the highest levels dictate further questions.

Will the data be stored as files on a traditional file system or in a database or in a yet another way? Will the whole data set be contained on one physical storage device or will it be distributed over several? How will the audio material be divided into individual files and directories? What additional metadata will be maintained [1]? In the following chapters, I shall list the original structuring for the Spoken corpus of Karel Makoň, provide reasons that led to that structuring, explain why a change was necessary, and describe the journey to realizing it.

2 The Spoken Corpus of Karel Makoň

The corpus arose as a set of amateur recordings on magnetic tapes. The author of the talks, Mr. Karel Makoň [2], a Czech mystic, started giving talks in private groups after

his deportation into a concentration camp in 1939, and stopped in 1991, two years before his death. His regular listeners were recording his talks. I don't know exactly when the recording started but the earliest denoted year is 1973. The earliest year I could make out from analyzing the contents is 1970 on an undated recording. Makoň had groups of attendants in various places of then Czechoslovakia.

There was a group in Pilsen, a group in Prague and a group in Gottwaldov, today's Zlín. Some of the recordings are from what we would now call retreats – meetings in a group in detached places that lasted several days. One such place was in Kaly near Brno in south Moravia. Another was in the cottage Čeřínek in the Bohemian-Moravian Highlands. Yet another was in Žiar, Slovakia. Other recordings are mostly from meetings hosted by one of the members.

2.1 Corpus Content

The talks can be seen as accompaniment to the author's written opus. Karel Makoň wrote twenty seven books on his own and translated and commented twenty eight books of other authors. Since the culmination of his spiritual path in the concentration camp in Sachsenhausen, where he acquired profound understanding of symbolism in Christianism and other traditions, he kept translating his experience to words in order to enable others to find enlightenment. His work can be compared to that of other modern spiritual masters, though there are pecularities.

Makoň bases his teachings mainly on Christian mystic, drawing heavily from the catholic tradition. He repeatedly uses the parable of the prodigal son and the parable of the talents in his arguments. He stresses that the words of Jesus must always be considered within the context of his deeds and always as a whole. He refers very often to Saint Teresa of Ávila, Augustine of Hippo and Padre Pio among others. He criticizes the catholic tradition for some points that he claims form an obstacle on the path to God and are in conflict with Jesus' intent.

One such example is the belief that the eternal life can only be reached after the physical death and that a life of virtues leads to it. Makoň claims that the eternal life – being consciously eternal – can only be reached within the physical body and that virtues alone never suffice to this end. He also draws from ancient Indian tradition, notably referring to Sat-Chit-Ananda as the three qualities of God. He translated and commented a book by Sri Aurobindo Ghos and draws from it. He also often refers to the life of Siddhartha Gautama Buddha, notably for his enlightenment in the moment of discarding the path of asceticism.

He suggests that the traditional hinduistic concept of re-incarnation should be revised firstly because a better understanding of what happens with the soul is at hand, secondly because, as he claims, correctly used Christianity leads to redemption within one incarnation. Aside from the symbolism in the New Testament, Makoň often uses milestones in his own life for explaining the general spiritual path. The most important of these milestones are repeated surgery without anaesthesia in the age of six, and confrontation with his own certain death in the concentration camp.

2.2 Original Recording

The vast majority of the recordings have been taken by one of the members of the Gottwaldov group. Fortunately, these recordings were taken using the best amateur technology available at that time and stored very carefully. Recordings from other places also exist but rather as rarities, with inferior acoustic quality and lacking systematic identification. A part of the recordings (30% of total length) was taken onto reels, the rest onto cassettes. Mostly, there was an identifier for a reel tape or for a cassette. Some pieces were unlabeled and some shared a label.

The majority of the recordings are cassettes with identifiers of the format `YY-NN`, for example `85-05` is the fifth recording taken in the year 1985. These occupy 686 resulting files out of total 802 files originating in cassettes. Of the 39 self-digitized reel-to-reel tapes, 36 were recorded at 9.53 [cm per second]. The remaining three in 2.38 [cm per second]. The latter took six hours to play back in one direction and the quality was heavily impaired. Of the reel-to-reel tapes, 24 are identified by a letter.

The sequence is roughly alphabetical, though there are three distinct ones identified by the letter I, and there is none with the letter G, which could have been lost. 85 are identified with year spanning from 1973 to 1988. There are may duplicities, however, so the actual number of distinct recordings is likely much lower in this category. 10 have a numerical identifier, 29 have a textual identifier, and 2 have no identifier whatsoever. There is also a three-hour video on youtube [1].

3 Digitization

For cassettes, the digitization has mostly been done one side of a cassette to one file. For reel-to-reel, it was one channel of one pass from reel to reel to one file. Notable exceptions are cassettes digitized in auto-reverse mode, which has been experimented with during the two years of the digitization. An exhaustive list of the volumes corresponding to each digitized file follows:

1. sides of cassettes: 615 files,
2. whole cassettes: 140 files,
3. reel-to-reel passes: 112 files,
4. imported, uncertain: 222 files,
5. two concatenated cassettes: 1 file.

The imported files are those that have been digitized by other parties prior to my own effort. The format for digitization was 48 [kHz], 16 [bit], real time. An exception to real-time digitization were the three reels recorded in 2.38 [cm/s]. These have been digitized in the standard speed of 9.53 [cm/s] and had the sample rate set to one quarter of the nominal value.

[1] https://www.youtube.com/watch?v=UaNm9jnnJiA

4 Usage of Digitized Material

The digitized audio files have been used for two purposes. On one hand for direct distribution over physical media and on the other hand in a dedicated web application that has been serving for direct playback in the browser and for acquisition of manual corrections to ASR-generated [3] transcription. There were two generations of said web application, see Krůza 2012 [4] for the first one, and Krůza 2018 [5] for the second one. The original version used the HTML audio element for playback.

This suffers from poor timing precision when playing back specific words but it handles streaming very well. The next generation of the web application uses the Web Audio API, which remedies the precision issues at the cost of streaming capability. Initially, the second generation web application would load the whole recording, decode it in-memory and only then enable playback and other operation.

Since 90 minutes is a very common length of a recording (264 of the total 1090 files are over 80 minutes long), the load on the user computer was very heavy. A raw, decoded 90-minute monaural recording in 24KHz occupies about 240MB. However, after decoding a recording like that, the browser would often occupy triple that amount of memory. Even when the user's computer could handle that, the waiting time was in order of minutes, which ensured a very poor user experience. To remedy this issue, a change was necessary.

5 New Structuring

To enable the web application to access random segments of a long recording, while keeping the precision provided by Web Audio API, these options emerge:

1. pre-load and decode the whole recording,
2. serve ranges by cutting the audio on the server side,
3. store the recordings in smaller chunks,
4. wait for streaming to be supported.

Option 1 was our baseline as discussed above. Option 2 - serving arbitrary ranges cut on demand by the server seems quite viable. It is flexible and not very hard to implement. One downside is that the server must be able to script as well as access the whole recordings. Since the amount is in order of tens of gigabytes even using compressed audio formats, itlimits the options for hosting and increases costs. Another downside is with caching. Repeated sessions with the same recording would likely lead to different ranges being requested, thus disabling the advantages of caching.

Lastly, computation-heavy operations on the server side can impair site reliability or require costly cloud solutions. Option 3 - storing the corpus in smaller chunks has the downside of serving unnecessary context when requesting a specific range. Also, if we want to retain full download capability, we must either stitch the recording together or host both split and integral versions of all recordings. Option 4 - waiting for streaming support in Web Audio API might seem silly but the issue has been discussed by the Web Audio API developer team[2].

[2] https://github.com/WebAudio/web-audio-api/issues/1305

I opted for storing the corpus in smaller chunks. The downside of unnecessary context is a minor one, especially if we choose a fitting length of the chunks. Having to host two versions of the corpus raises the costs in my case by less than one US Dollar per month.

6 The Process of Restructuring

There are four choices to be made for the restructuring:

1. how long will the chunks be,
2. how to choose individual split points,
3. what will be the directory structure,
4. the file names.

6.1 Segment Length

The length of the chunks should be no more than 5 minutes. With 24kHz compressed audio, a 5-minute chunk takes up about 1.5MB. Theoretically, with modern 3G connections, even larger downloads should be instantaneous. However, the reality diverts from table speeds and we also have to wait for decoding, which takes about 4 seconds for a 5-minute chunk with an Intel Core2 @ 2.5GHz. According to the magazine UXMovement [8], four seconds are the threshold above which the user starts to abandon the original intent.

The Nielsen Norman Group [6] claims the same for ten seconds. The lower boundary is much freer but there always is a chance of an artifact at the glue point during playback, so the less of them the better. Also, each chunk means an extra HTTP request, which itself has considerable overhead. A reasonable compromise seems to be a span of 30 to 120 seconds for a chunk.

6.2 Finding Split Points

By choosing split points well, the impact of occasional artifacts caused by imprecisely switching segments during playback can be reduced. Ideally, splitting should be performed in pauses between sentences or at least between words. It is practically impossible to speak for two minutes without taking a breath, so there should always be a pause to be found for a segment's boundaries. There are numerous ways to find a moment of silence in an audio recording. The most requiring and most reliable way is manual annotation.

Another reliable method is by looking for predicted silences in phone-level-aligned transcription. This method could largely be used because there are manual or automatically-acquired transcriptions to most of the recordings in the corpus. Where either speech recognition or its forced alignment failed, and thus there is no transcription, another method comes to question: Voice-activity detection (VAD). I have used the perl module Audio::FindChunks for this purpose.

Table 1. Number of split points by their method of acquisition

acquisition method	number of uses
manually	0
by aligned transcription	60424
by voice activity detection	0
fix size	22043
total	82467

This method, unfortunately, is not very reliable in case of audio input with low signal-to-noise ratio, which is an abounding phenomenon in the Spoken Corpus of Karel Makoň. Where not even VAD helps, which can be identified so that the detected chunks are too long, there only remains the blunt method of creating fixed-size segments, disregarding the frequent case of splitting inside of a word.

Two of these methods have been eliminated by experiments: Manual annotation was way too inefficient. Six volunteers attempted this task, losing their patience after zero to ten minutes of transcribed material. Furthermore, detection by aligned transcription could be used for some recordings where initially speech recognition failed. By splitting the recordings into smaller chunks of fixed size and then recognizing each segment apart, the problematic recordings could be speech-recognized.

The resulting accuracy was catastrophic given the bad acoustic quality of these recordings but the longer silences were mostly precisely detected[3]. The chunks where the ASR failed again, were also beyond what VAD could handle, so there was no reason to resort to it. Table 1 summarizes the number of split points acquired by each method. The significant number of split points acquired by fix-size spliting is due to the fact that there are still recordings for which the split-and-recognize procedure is yet to be performed.

6.3 Split Point Selection

For fix-size segment splitting, I chose a segment length of 60 seconds. For splitting by aligned transcription, the task at hand is one where the input is a sequence of silences in the recordings, each identified by start and end, and the output is an optimal subsequence of these silences. The weight used is length of the silence (the longer the pause, the better it is as a split point). A constraint is the distance of the selected silences to be between 30 and 120 seconds.

Notice the task has no solution if the recording is shorter than 30 seconds. However, this never occurs in the Spoken Corpus of Karel Makoň and even if it did, we could simply leave the short recording in one segment. Another case where there is no solution is when the distance between adjacent silences (strictly speaking between the mid-points of adjacent silences) is greater than 120 seconds. Such a case occurs when the silences themselves are very long.

[3] There are no gold standard data, so there's no way of evaluating the result exactly but so far, I found no split in the middle of a word using this technique.

I have solved such cases manually but an automated approach would also be very simple. The algorithm sought seems to be a typical demonstration of dynamic programming: Find an optimal split of a part of the recording that only consists of the first word, then add next word and find the optimal split based on the current one. There is also a simpler variant, however, this algorithm is very easy to program and it has linear complexity.

1. start with the set of all silences;
2. iterate over them from shortest to longest;
3. remove the silence from the set if joining its adjacent segments doesn't yield one that's longer than 60 seconds;
4. end iteration;
5. iterate again over chosen silences;
6. remove a silence if one of its adjacent segments is shorter than 30 s;
7. end iteration.

6.4 File Naming

Naming of the chunks is arbitrary as long as the file names are unique. But a good naming scheme can help with further processing of the data. A good naming scheme can be kept when the split points are changed for a recording, without collisions. This excludes simple ordinal numbering. Another criterion is ease of filtering and parsing of the file names. Lastly, the if the file name is descriptive and human-readable, it can save time and effort. To accommodate these criteria. I chose to name the chunks as follows: `recording-identifier--from-start-time--to-end-time.format`

For example `87-25B--from-2186.45--to-2239.63.ogg`. To ensure smooth transitions, the segments are encoded 500 milliseconds longer than declared. This is to compensate for rounding of the length of the split file when encoded in a compressed format. Of course, this commands for the algorithm used in the web application to switch playback from one segment to another before the actual last sample of the expiring segment.

7 Reproduction

The corpus can be accessed through its web interface at[4]. The playback and other operations like playing back or downloading an arbitrary span (which can span a segment boundary) can be tested there. The source code for splitting the audio can be accessed at GitHub [5]. Similarly, the source code for the JavaScript part that works with the segments can be found at its GitHub repository [6], specifically in the files `src/store/audio.js` and `src/store/AudioChunks.js`.

[4] http://radio.makon.cz/

[5] https://github.com/Sixtease/CorpusMakoni/tree/master/scripts

[6] https://github.com/Sixtease/MakonReact

8 Conclusion

The new segmented structuring is used in the web application to emulate streaming capability while exploiting the advantages offered by Web Audio API. Other uses of the corpus, like direct downloads, archiving and on-demand forced alignment still rely on the original structuring where one recording of length in order of tens of minutes corresponds to a single file.

The redundant storage needs impose a negligible financial overhead but the work necessary for creation and maintenance of these extra files bring about an even stronger wish for the Web Audio API to support streaming. However, storing the corpus in files of a smaller size and devising the mechanism to seamlessly glue the chunks together during playback and transcription was a worthy lesson with potential use in other settings.

Acknowledgments. This work has been using language resources developed, stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2010013). Participation in the conference was supported by COST action IS1404. This research was partially supported by SVV project number 260 104.

References

1. Crowdy, S.: Spoken corpus design. Literary and Linguistic Computing, vol. 8, no. 4, pp. 259–265 (1993)
2. Hájek, J.: Český mystik Karel Makoň. Dingir, vol. 2007/4, pp. 142–143 (2007)
3. Ircing, P., Krbec, P., Hajic, J., Psutka, J., Khudanpur, S., Jelinek, F., Byrne, W.: On large vocabulary continuous speech recognition of highly inflectional language-Czech. Seventh European Conference on Speech Communication and Technology, (2001)
4. Krůza, O., Peterek, N.: Making community and ASR join forces in web environment. International Conference on Text, Speech and Dialogue. Springer, pp. 415–421 (2012)
5. Krůza, O., Kuboň, V.: Second-generation web interface to correcting ASR output. Proceedings of the Future Technologies Conference (FTC). Science and Information Organization. Springer-Verlag, vol. 1, no. 1, pp. 749–762 (2018)
6. Nielsen, J.: Website response times (2010)
7. Reppen, R.: Building a corpus: what are the key considerations? The Routledge handbook of corpus linguistics. Routledge, pp. 59–65 (2010)
8. Tseng, A.: Progress bars vs. spinners: When to use which (2016)