Validation of Speech Data for Training Automatic Speech Recognition Systems

Janez Križaj
Faculty of Electrical Engineering
University of Ljubljana
Ljubljana, Slovenia
0000-0001-9581-2615

Jerneja Žganec Gros
Alpineon d.o.o.
Ljubljana, Slovenia
0000-0001-5011-8486

Simon Dobrišek
Faculty of Electrical Engineering
University of Ljubljana
Ljubljana, Slovenia
0000-0002-9130-0345

***Abstract*—Recent automatic speech recognition systems are largely based on deep neural networks that need large amounts of labelled speech data to train. This can be a problem, especially for languages for which large speech databases are not available. To facilitate the construction of a speech database suitable for training automatic speech recognizers, we propose a tool that enables the validation of audio recordings from collected speech recordings. The developed tool allows the user to check the compliance with the predefined requirements regarding the correct audio format, the appropriate speech volume, the compatibility of the spoken text with the reference text and the suitability of the length of the non-spoken segments. The applicability of the developed tool is demonstrated by the creation of the Slovene speech corpus from audio recordings collected within the project Development of Slovene in a Digital Environment, although the tool is also suitable for all other languages supported by the used automatic speech recognizer.**

***Index Terms*—speech corpora, automatic speech recognition, audio validation**

I. Introduction

The field of automatic speech recognition (ASR) has made great progress in recent years, mainly due to the widespread use of deep neural networks [1], [2], which, however, require a lot of training material for their learning. Unfortunately, there are currently no suitable freely available speech databases for the Slovene language that could be used to build a speech recognizer for Slovene and commercial products derived from it, or the existing databases are not large enough to build a recognizer according to modern standards. As part of the Development of Slovene in a Digital Environment (DSDE) project [3], our goal is to create a Slovene speech database that is freely accessible and can be used for both non-commercial and commercial purposes.

In order to successfully train an ASR system, the training data should meet certain predefined requirements, which relate in particular to the sufficiently high quality of the audio recordings and the compliance of the recorded speech with the reference text. The tool presented in this article makes it possible to check compliance with the aforementioned requirements and to reject recordings that do not meet these requirements and could therefore negatively affect the machine learning of the recognizer. The main purpose of the implemented tool is to facilitate the validation of audio recordings for the construction of the speech database within the DSDE project. The speech database will enable the development of advanced Slovenian ASR systems that is not possible with currently available resources and tools. The current speech databases of the Slovenian language allow us to build a speech recognizer with Word Error Rate (WER) between 25 % and 30 % and limited to the acoustic situations covered in these databases [4]. We expect to significantly improve this WER with the new speech database.

The work presented in this paper includes the following contributions:

1) Development of a tool for the validation of audio recordings, that allows the elimination of recordings that do not meet the predefined requirements.
2) Use of the developed tool for the construction of a speech database, which will be the basis for the creation of a speech recognizer of Slovene.

The following sections provide a detailed overview of the developed tool and the preliminary results obtained from the speech recordings collected during the DSDE project.

II. Audio Validation Tool

This chapter provides an outline of the main building blocks and their role in the developed tool for validating speech recordings.

A. The Process of Collecting Audio Recordings

In general, there are two approaches to building speech databases. In the first approach, individual sentences of text read by speakers are recorded, with each recording containing only one sentence. In the second approach, a longer audio file and the corresponding text are subsequently broken down into individual sentences [5]. To ensure accurate alignment of the speech with the reference text, we focus on the first approach.

Each speaker is given a list of sentences previously collected through web scraping of online news portals in Slovenian language [6]. When recording audio, the speaker should adhere to the prescribed requirements: *i*) each individual sentence should be saved in a separate audio file with the WAV extension, *ii*) there should be at least half a second and no more than one second of the originally recorded silent pause at the beginning and the end of each sentence, *iii*) the recordings must be originally recorded and saved in a single-channel format (mono) and the sampling frequency of $44.1\,\mathrm{kHz}$.

B. Validation Process Description

The developed tool facilitates the validation of the mentioned requirements that should be considered during audio recording. A flowchart of the validation process is shown in Fig. 1, which shows that compliance with the prescribed requirements is validated through three consecutive assessments that include audio format verification, matching between the spoken text and the reference text, adequacy of initial/final silence lengths and audio volume. If the checked audio does not meet any of the three conditions, it is added to the list of rejected recordings, otherwise it is added to the list of accepted recordings.

C. Graphical User Interface

User interaction with the proposed tool is possible through the graphical user interface (GUI) shown in Fig. 2. The interface window is divided into several frames. In the upper left corner there is a frame for entering input arguments, the lower left frame is for evaluating the correspondence between the spoken text and the reference text, while the graphs in the middle frame are for evaluating the initial and final silence lengths and audio volume. On the right side there is a frame that displays statistical data of the processed audio file.

To start the validation process, first the input parameters should be entered. These include *i*) the directory path containing the WAV audio files, *ii*) the path to the XLSX file containing the reference text, *iii*) the number of the audio track where we want to proceed with the validation process, *iv*) the operating mode, where we choose among

1) *automatic*, where the verification of all three conditions is performed automatically and the program does not require any user interaction after the input parameters have been entered;
2) *semiautomatic* which requires user intervention only if any of the three conditions are not met. In this case, the GUI allows the user to manually reject the audio if, after the manual review, the user believes that the condition in question is not met.
3) *manual* where the user responds to each of the three conditions by pressing the corresponding button in the GUI.

If the user selects the automatic or semiautomatic mode, it is necessary to enter a threshold value WER explained in the next subsection, above which it is assumed that the spoken text does not match the reference text. After entering the input parameters, the validation is started by pressing the "Run" key.

D. Reference Text Matching

The validation tool has two approaches to verify that the audio recording matches the reference text. In manual mode, the audio clip is played in the GUI and the reference text is displayed in a special frame, allowing the user to judge the match and accept/reject the audio clip by pressing the corresponding button. In (semi)automatic mode, the match with the reference text is (semi)automatically assessed using Google's ASR engine [7]. The engine is based on recurrent neural networks and operates as a cloud service. It can be used with the default credentials without the need to log in to the cloud service.

The text string returned by the automatic speech recognizer is then compared to the reference text using WER (Word Error Rate), a commonly used metric for evaluating the quality of automatic speech recognizers [8], and is defined as

$$\mathrm{WER} = \frac{S + D + I}{N}\,, \tag{1}$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions, and N is the number of words in the reference text. Since all numbers in the reference text are written in words, each number in the Google Automatic Speech Recognizer output is converted to words before the WER is computed using (1). Since the speech recognition output does not contain punctuation, it is also removed from the reference text before computing WER.

In (semi)automatic mode, the match with the reference text is confirmed/rejected if the value of WER is below/above the preset threshold. In manual mode and in case of rejection in semiautomatic mode, the match of the audio with the reference text is checked by the user listening to the recording. Differences between the reference text and the recognized text are color coded, with substitutions in orange, insertions in green and deletions in red (see lower left frame in Fig. 2).

E. Silence Length and Audio Volume Analysis

The length of the initial/final non-speech segments and the audio volume are also estimated either automatically or manually, depending on the selected operating mode. For automatic estimation of non-speech lengths, we used the Voice Activity Detector (VAD) based on Gaussian Mixture Models and developed by Google [9]. To discard potential short interpauses within the speech segment, a set of heuristic parameters is added to the estimation process.

In manual mode of operation, the GUI displays three graphs (middle frame in Fig. 2) to help decide the validity of initial/final silence lengths and audio volume. The top graph shows the amplitude of the audio clip, the middle graph shows the spectrogram and the bottom graph shows the loudness of the audio. The amplitude and loudness graphs have automatically calculated non-speech (in red) and speech (in green) sections color-coded, making it easier for the user to check the suitability of the initial/final pauses. The dashed horizontal lines on the loudness plot delimit the desired range of speech loudness, which is between $-18\,\mathrm{dBFS}$ and $-6\,\mathrm{dBFS}$. The spectrogram plot in the middle helps to check if the initial/final silence was artificially created and added to the audio at a later time, which is also a reason for rejecting the recording.

F. Implementation Details

The tool is implemented in Python 3. Most tasks in the audio validation process are implemented using existing program-

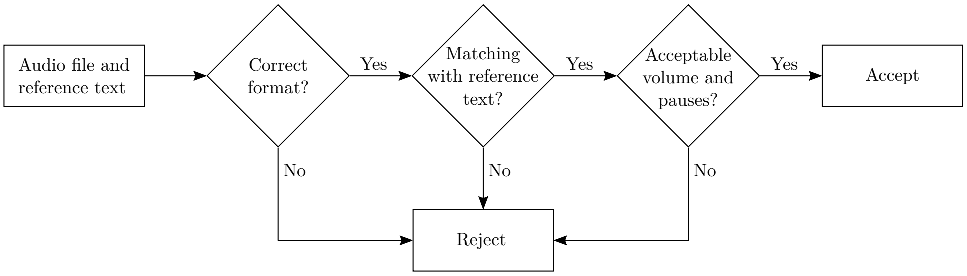
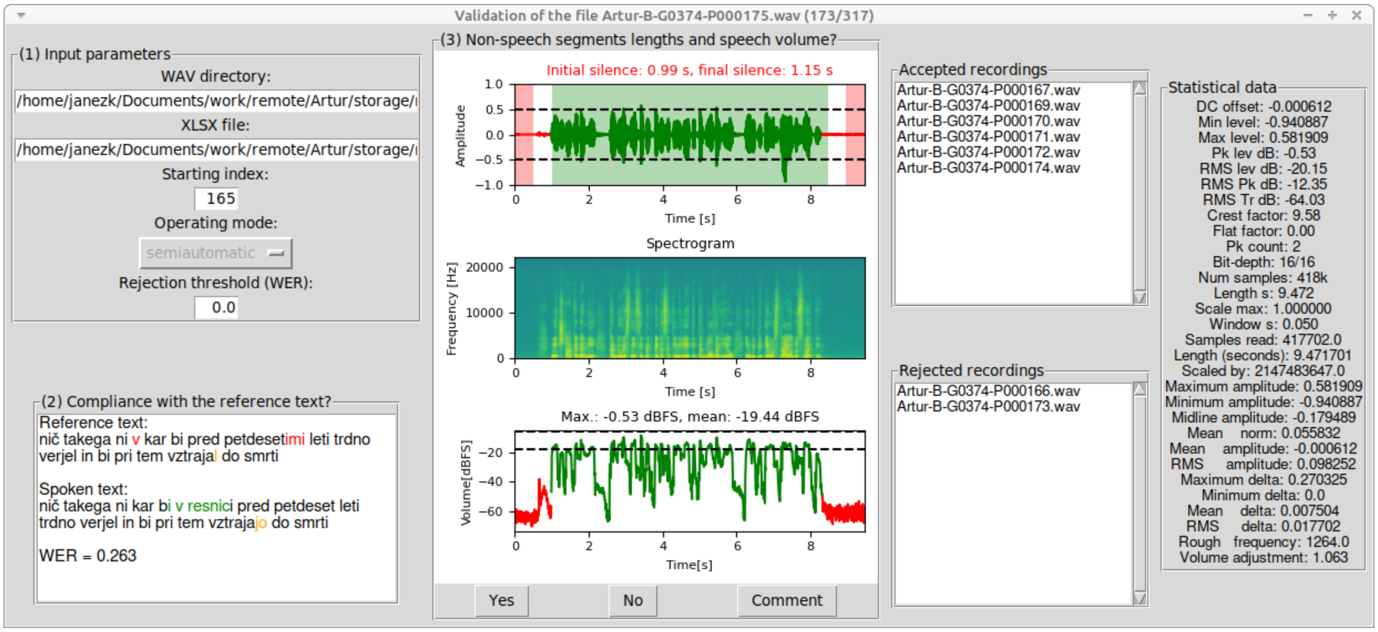

Audio file and reference text
Correct format?
Yes
Matching with reference text?
Yes
Acceptable volume and pauses?
Yes
Accept
No
No
No
Reject

Fig. 1. Schematic representation of the audio validation process. The input audio clip is validated by three successive tests that check the audio format, the compliance with the reference text and the suitability of the initial/final non-speech segments along with the volume of the audio.


Validation of the file Artur-B-G0374-P000175.wav (173/317)
(3) Non-speech segments lengths and speech volume?
Initial silence: 0.99 s, final silence: 1.15 s
Amplitude
1.0
0.5
0.0
−0.5
−1.0
0
2
4
6
8
Time [s]
Spectrogram
Frequency [Hz]
20000
10000
0
0
2
4
6
8
Time [s]
Max.: -0.53 dBFS, mean: -19.44 dBFS
Volume[dBFS]
−20
−40
−60
0
2
4
6
8
Time[s]
Yes
No
Comment

(1) Input parameters

WAV directory:
/home/janezk/Documents/work/remote/Artur/storage/i

XLSX file:
/home/janezk/Documents/work/remote/Artur/storage/i

Starting index:
165

Operating mode:
semiautomatic

Rejection threshold (WER):
0.0

(2) Compliance with the reference text?

Reference text:
nič takega ni v kar bi pred petdesetimi leti trdno
verjel in bi pri tem vztrajal do smrti

Spoken text:
nič takega ni kar bi v resnici pred petdeset leti
trdno verjel in bi pri tem vztrajajo do smrti

WER = 0.263

Accepted recordings

Artur-B-G0374-P000167.wav
Artur-B-G0374-P000169.wav
Artur-B-G0374-P000170.wav
Artur-B-G0374-P000171.wav
Artur-B-G0374-P000172.wav
Artur-B-G0374-P000174.wav

Rejected recordings

Artur-B-G0374-P000166.wav
Artur-B-G0374-P000173.wav

Statistical data

DC offset: -0.000612
Min level: -0.940887
Max level: 0.581909
Pk lev dB: -0.53
RMS lev dB: -20.15
RMS Pk dB: -12.35
RMS Tr dB: -64.03
Crest factor: 9.58
Flat factor: 0.00
Pk count: 2
Bit-depth: 16/16
Num samples: 418k
Length s: 9.472
Scale max: 1.000000
Window s: 0.050
Samples read: 417702.0
Length (seconds): 9.471701
Scaled by: 2147483647.0
Maximum amplitude: 0.581909
Minimum amplitude: -0.940887
Midline amplitude: -0.179489
Mean norm: 0.055832
Mean amplitude: -0.000612
RMS amplitude: 0.098252
Maximum delta: 0.270325
Minimum delta: 0.0
Mean delta: 0.007504
RMS delta: 0.017702
Rough frequency: 1264.0
Volume adjustment: 1.063

Fig. 2. Graphical user interface of the proposed validation tool. The upper left frame is used for input parameters, the lower left frame is used for matching with the reference text, the middle frame displays the non-speech section and the audio volume evaluation, while the frames on the right display lists of accepted and rejected audio clips and statistical data.

ming libraries listed in Table I. The software is freely available at https://github.com/jan3zk/audio_validation. In addition to the Python scripts, there are also standalone executables for Windows and Linux that we created using the PyInstaller library and can run the tool without having to install the Python libraries.

III. Experiments

The proposed audio validation tool was tested on speech recordings collected within the DSDE project with the aim of building a speech corpus for training automatic speech recognition systems of Slovene.

A. Error Analysis

In the first set of experiments, we investigate the reasons for audio clip rejections and derive (semi)automatic validation errors by considering the deviations from rejections in manual mode, which we consider error-free. The results we have obtained with the audio collected so far are shown in Table II.

The rejection rate due to an incorrect audio format does not depend on the selected operating mode, since the format is automatically checked in all modes.

The rejection rate due to mismatch with the reference text in manual mode is comparable to that in semiautomatic mode. In both cases, the text mismatch rejection rates are around 1/10 of all validated audio clips. These rejection rates are much higher in the automatic mode of operation, as almost 2/3 of the audio clips are rejected when the threshold is set to WER $= 0.0$. The increase in rejections due to mismatch with reference text in automatic mode can be attributed to false rejections due to errors in automatic speech recognition. In semiautomatic mode, where each rejection is manually checked, such false rejections can be rolled-back by the validator.

The rejections due to inadequate length of initial/final silences and/or audio volume are less than 1/10 of all validations, regardless of the operating mode used. This type of rejections is only slightly higher in automatic mode, from which we can conclude that automatic estimation of non-

TABLE I
UTILIZED PROGRAMMING LIBRARIES

Task	Library
GUI	tkinter
audio format validation	soundfile
reference text matching	Google STT API, SpeechRecognition, difflib, jiwer, num2words
silence length and audio volume estimation	scipy, pydub, matplotlib, sox
conversion from .py to .exe	pyinstaller
acoustic normalization	noisereduce, SpeechDenoisingWithDeepFeatureLosses, santi-pdp/segan
speech quality estimation	speechmetrics

TABLE II
REJECTION RATES BY CAUSE

	Rejection reason		
Operating mode	format*	text†	pause/vol.§
manual	0.001	0.091	0.078
semiautomatic	0.001	0.098‡	0.079
automatic	0.001	0.624‡	0.090

* incorrect format
† mismatch with the reference text
§ unsuitable non-speech length or audio volume
‡ rate at WER = 0.0

speech segments and audio volume is quite reliable. Examples of automatic estimation of non-speech segments from noisy audio can be found in Fig. 3, where segments are reliably estimated even when the noise volume in non-speech segments sometimes exceeds the speech volume.

The results in the Table II show that due to the insufficient reliability of automatic speech recognition, it is advisable to choose a semiautomatic operating mode that ensures low number of automatic false acceptances at low WER values, while allowing manual filtering of false rejections, thus achieving faster validation and accuracy comparable to the manual mode.

Table III shows the errors of the validation procedure in terms of false acceptance rates (FAR) and false rejection rates (FRR). The manual mode of operation serves as the ground truth for the other two modes. As mentioned earlier, the automatic mode is the most error-prone, where nearly 2/3 of the audio data is falsely rejected due to errors in the ASR module. In the semiautomatic mode, the false rejections caused by the inaccurate ASR are manually reversed. Thus, the error rates in semiautomatic mode primarily relate to inaccurate automatic estimation of non-speech segments.

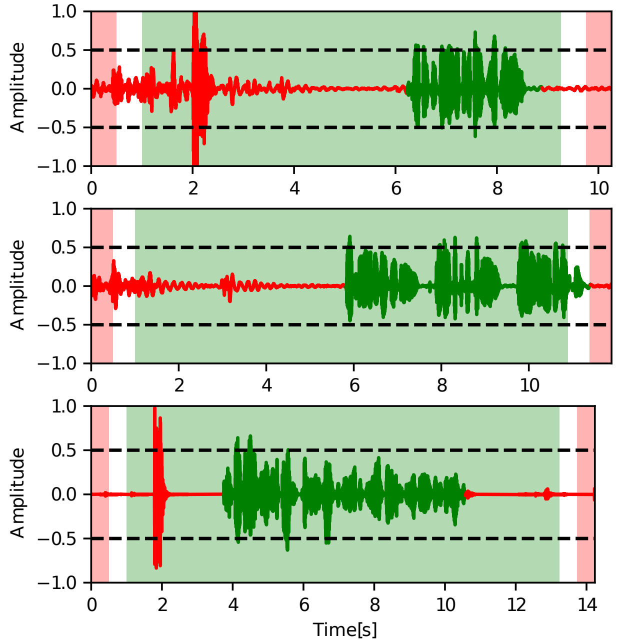


Fig. 3. Examples of automatically estimated non-speech sections in the noisy audio clips. The green part indicates detected speech sections, while the red part covers initial and final silence. Note that silent sections are reliably detected despite the large amount of noise that usually comes from background clutter or coughing.

TABLE III
VALIDATION ERRORS AT WER = 0.0

	Error rate	
Operating mode	FAR	FRR
manual	0.00	0.00
semiautomatic	0.02	0.01
automatic	0.02	0.53

B. Time Requirements

The time required to validate an audio clip depends heavily on the choice of operating mode. The time values in the Table IV show the average time required to validate a single audio recording and were calculated using a subset of 100 recordings. Automatic mode is the fastest and requires no user intervention, but is also prone to validation errors, as shown in the previous section. Manual mode takes the longest, since each recording requires user intervention. The speed of validation in semiautomatic mode depends on how many of the recordings are automatically accepted. If the threshold is set to WER = 0.0, almost 2/3 of the recordings still have to be validated manually, which increases the validation time accordingly.

IV. CONCLUSION

The paper presents a speech validation tool developed to facilitate the verification of compliance with the specified requirements during the creation of a speech corpus for the training of automatic speech recognition systems. The speech

TABLE IV
AVERAGE TIME REQUIREMENTS TO VALIDATE A SINGLE AUDIO RECORDING

Operating mode	Time requirements [s]
manual	17.1
semiautomatic	11.9*
automatic	4.7

* at WER = 0.0

validation tool allows the user that the recorded speech data meets the predefined requirements in terms of a specific audio format, appropriate speech volume, the compliance of the spoken text with the reference text in the case of read speech, and the appropriateness of the start and end segments outside the speech. The effectiveness of the tool was demonstrated in the use case of acquiring a Slovene speech corpus from audio recordings collected in the DSDE project.

Future work includes the use of various automatic speech recognition systems to verify that the recorded text matches the reference text. We also plan to evaluate how the accuracy of various ASR systems trained on the newly collected speech corpus compares to those trained on existing freely available databases of Slovene, such as Mozilla Common Voice [10] and VoxPopuli [11]. Our goal is to develop a general and two specialized speech recognizers in the form of a freely accessible cloud service that allows users to create a transcription for an uploaded audio file.

ACKNOWLEDGMENT

The research that led to the study presented here was funded in part by operation Development of Slovene in the Digital Environment, which is co-financed by the Slovenian Ministry of Culture and the European Regional Development Fund.

REFERENCES

[1] S. Kriman, S. Beliaev, B. Ginsburg, J. Huang, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li, and Y. Zhang, "Quartznet: Deep Automatic Speech Recognition with 1D Time-Channel Separable Convolutions," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6124–6128.

[2] Y. Kong, J. Wu, Q. Wang, P. Gao, W. Zhuang, Y. Wang, and L. Xie, "Multi-Channel Automatic Speech Recognition Using Deep Complex Unet," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, 2021, pp. 104–110.

[3] "Development of Slovene in a Digital Environment," https://slovenscina.eu/en/about-project, accessed: 2022-02-28.

[4] M. Ulčar, S. Dobrišek, and M. Robnik-Šikonja, "Automatic slovene speech recognition using deep neural networks," *Applied Informatics*, vol. 27, no. 3, Sep. 2019. [Online]. Available: https://uporabna-informatika.si/index.php/ui/article/view/53

[5] E. Bakhturina, V. Lavrukhin, and B. Ginsburg, "A Toolbox for Construction and Analysis of Speech Datasets," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[6] J. Žganec Gros, B. Vesnicer, and S. Dobrišek, "A Method for Selection of Phonetically Balanced Sentences in Read Speech Corpus Design," in *Proceedings of the 30th European Signal Processing Conference, EUSIPCO*, 2022, "Manuscript submitted for publication".

[7] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.

[8] A. Ali, W. Magdy, P. Bell, and S. Renais, "Multi-reference WER for evaluating ASR for languages with no orthographic rules," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, pp. 576–580.

[9] "Google WebRTC VAD," https://webrtc.org, accessed: 2022-01-10.

[10] R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber, "Common Voice: A Massively-Multilingual Speech Corpus," in *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 4218–4222.

[11] C. Wang, M. Riviere, A. Lee, A. Wu, C. Talnikar, D. Haziza, M. Williamson, J. Pino, and E. Dupoux, "VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, aug 2021, pp. 993–1003. [Online]. Available: https://aclanthology.org/2021.acl-long.80