Unsupervised Perturbation-based Quality Estimation for Speech-to-Text Translation Systems

Bachelor's Thesis of

Iva Andreeva

Artificial Intelligence for Language Technologies (AI4LT) Lab
Institute for Anthropomatics and Robotics (IAR)
KIT Department of Informatics

Reviewer:	Prof. Dr. Jan Niehues
Second Reviewer:	Prof. Dr.-Ing. Tamim Asfour
Advisor:	M.Sc. Tu Anh Dinh

20. May 2024 – 20. September 2024

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

PLACE, DATE

. . . . . .

(Iva Andreeva)

Abstract

As the use of Spoken Language Translation systems grows more prevalent, it is accompanied by a staggering lack of quality estimation methods for their generated translations. Of the few that exist, none are unsupervised and none are applicable to a black-box scenario. This work's product is a novel quality estimation method which provides reference-free quality estimates without using system-specific information and without needing training or training data, but instead relying on the perturbation of the source language audio. The audio perturbation framework we built in scope of this work includes frequency band filtering, noising, resampling and speed-pitch-warping. We evaluate our method in correlation with reference-based COMET quality assessments and achieve almost five times higher performance than the baseline of output sequence probabilities of the translating model.

Zusammenfassung

Mit dem steigenden Wachstum der Anwendungsfälle von Sprach-zu-Text Übersetzungssystemen wird ein überwältigender Mangel an Qualitätsschätzungsmethoden für deren generierte Übersetzungen erkenntlich. Von den wenigen existierenden Methoden mit diesem Ziel sind keine unüberwacht und keine anwendbar auf ein Black-Box-Szenario. Das Produkt dieser Arbeit ist eine neuartige Qualitätsschätzungsmethode, die ohne die Verwendung von Referenzübersetzungen, ohne systemspezifische Informationen und ohne Training oder Trainingsdaten zurechtkommt und sich stattdessen auf die gezielte Veränderung des Originalsprachenaudios verlässt. Das im Rahmen dieser Arbeit entstandene Framework zur gezielten Audiomanipulation umfasst Frequenzbandfilter, Berauschung, Umsampling und Geschwindigkeits- und Tonhöhenverschiebung. Wir evaluieren unsere Methode in Korrelation mit den referenzbasierten Qualitätsvoraussagen von COMET und erreichen nahezu fünffach höhere Ergebnisse im Vergleich zur Ausgabesequenzwahrscheinlichkeit des Übersetzungsmodells.

Contents

List of Figures

List of Tables

1. Introduction

Considering the vastly increased capability of language translation models in recent years, it is only natural to observe an increase in demand for their use. Accompanying this rise is the diversification of their domains of application, which makes reliance on translation systems for matters of lives and livelihoods more common. Judicial and medical decisions based on foreign-language information, for example, can be laden with grave implications for those involved. In such domains, translation quality holds vital importance, but it is not always viable or even possible to have it guaranteed by a certified translator or even a native speaker. Therefore, the automated estimation of said quality without using a reference translation becomes necessary.

Although many methods to estimate the quality of a given machine-generated Text-to-Text Translation have already been developed, there is a severe lack of such methods for Speech-to-Text Translation. The few that exist, proposed by Le et al. [11] and Besacier et al. [2], use system-specific information from the translation-generating system to produce their quality estimates. This glass-box approach forces their methods into being architecture-specific, making it difficult to apply them to certain translation systems, but also to use the method at all when working with generations from an inaccessible system whose interface does not allow sufficient access to the required information.
Notwithstanding their use of glass-box information, Le et al.'s [11] and Besacier et al.'s [2] presented works on the topic culminate in the supervised training of a classifier, introducing domain dependence into the resulting quality estimation method. Taking into context the aforementioned trend towards the diversification of translation system application domains, this introduced domain specificity of quality estimation methods seems untimely and calls for the development of unsupervised methods instead.

In this work, we present our new method Perturbation-based QuESTT: Perturbation-based Quality Estimation for Speech-to-Text Translation. To our knowledge, it is the first reference-free, black-box and unsupervised method for the quality estimation of Spoken Language Translations. By using variations on tried and tested audio augmentation methods as proposed by Nanni et al. [14], we let the translating system make predictions for the original, as well as multiple perturbed versions of the source language audio. Using common text similarity measures like BLEU [16], the Translation Edit Rate (TER) [25] and the Character n-gram F-score (CHRF) [18], we compute the variation present in the predicted translations of the perturbed audio compared to the prediction of the original. We use the translation's robustness and invariance under perturbation as an indicator of quality and make quality estimation predictions on their basis.

We perform an experimental evaluation of our method using multiple different perturbation and hyperparameter configurations on the IWSLT23 Quality Estimation dataset by

Sperber et al. [27]. We record very good results, achieving nearly five times higher Pearson correlation of our best-performing QE model variant with reference-based COMET scores than the translating model's output sequence probability baseline.

The implementation of our experiments can be found in our GitHub repository[1], including supplemental illustrative data on our various used perturbation strategies. More detailed insight into Quality Estimation and Perturbation can be found in our Section 2 on related work. We illustrate how we apply this knowledge to the development of Perturbation-based QuESTT in Section 3. Section 4 then describes the implementation of our experiments, laying the groundwork for our evaluation, which we document in Section 5. Based on our findings, we offer possibilities for future work and conclude our work in Section 6.

[1] https://github.com/13thWitch/QE-for-S2TT

2. Related Work

Quality Estimation (QE) aims to autonomously provide quality assessments for machine generated output without using an output reference [12] [28]. In Natural Language Processing (NLP) this assessment is made on word level, analogously to a sequence labelling problem [5], or on sentence level, commonly as a regression problem [22].
The machine-generated natural language sequences whose quality is meant to be estimated are often translations, which are the focus of this work.

2.1. Quality Estimation of Text Translation

The nature of data used to produce the translation quality assessments classify QE methods into *glass-box*, when utilizing system-specific information extracted before or during the translation, and *black-box*. Black-box QE allows for the generation of system-independent quality estimations without needing access to the generating systems' inner workings. Thus, it proves optimal for application in common use-cases of Machine Translation (MT) systems with limited access to the systems themselves, like proprietary translation systems.

Most state-of-the-art QE methods are supervised [7] [26], meaning that they are learned using human-labelled QE data which can be costly to obtain, erroneous, and can make the QE method susceptible to domain-dependence [8]. These negative factors contributed to the rise in development of unsupervised QE methods. Often combined with glass-box knowledge, approaches to unsupervised QE have been discussed and improved [15] [6] without being able to match the performance of supervised QE methods [7]. For example, Niehues and Pham [15] estimate quality by calculating similarity between test input and the examples seen in training, while Fomicheva et al. [6] use the posterior token probabilities and the learned attention weights, limiting the applicability of their method to attention-based models.

Mixed approaches attempt to avoid QE data sparsity by learning from synthetic data. One such approach by Tuan et al. [30] creates pseudo-labelled QE-data using an MT-system to produce dirty translations. In turn, this method requires large amounts of MT data, meaning source text and gold-standard translations, introducing not only MT-system dependence into the QE method, but also domain dependence.

Perturbation-based QE by Dinh and Niehues [5] avoids these problems by assuming a black-box scenario and eliminating training data as a necessity. They perform word-level quality estimations working under the assumption that a token in the generated translation is badly translated if it depends on too many parts of the source text. These word-level assessments can be aggregated to form a sentence-level quality estimation. The dependence of a translated token on specific parts of the input is extracted by variously

perturbing the source text and monitoring resulting changes in the generated translation. Therefore, only access to the generating model itself is required; none to its inner workings. The method developed in this work, Perturbation-based QuESTT, is based on this T2TT QE-method by Dinh and Niehues [5] and, as its counterpart for Speech-to-Text Tranlsation (S2TT), requires neither system-specific information, nor training data. We propose a novel method built around input perturbation to perform unsupervised, black-box QE for the S2TT Scenario.

2.2. Quality Estimation of Spoken Language Translation

When working with S2TT-models, two general architectures present themselves: The *cascade* approach, which employs an automatic speech recognition (ASR) component followed by T2TT, and direct *End-to-End* S2TT solutions which are able to retain prosody information and make for simpler training [21]. If working with cascade models and following a glass-box QE approach, using the transcription generated by the ASR component can easily be used to apply T2TT QE methods as indicated above.
For black-box QE the cascade and End-to-End models are equivalent from a method design perspective, as the inner structure and any differences therein are unusable and therefore irrelevant. However, a performance gap in favor of cascade models has been observed in most but for the newest End-to-End models like ZeroSwot by Tsiamas et al. [29] [21], which could challenge S2TT-QE method robustness. QE methods which are not sufficiently robust and evaluate more accurately on higher performance End-to-End models could prove to perform worse on cascade models, and vice versa.

Neither for cascade, nor for End-to-End architectures has S2TT QE received much attention from the NLP community. As far as we know, the only published quality estimation methods for S2TT were proposed by Besacier et al. in 2014 [2], followed by Le et al. in 2016 [11]. Aiming to create a Word Confidence Estimation System for cascade S2TT, they both use black- and glass-box methodology to train a supervised model and predict word-level quality estimates. To achieve this, they separate their evaluated system into its ASR and T2TT components, making their methods only applicable for cascade S2TT models.

To our knowledge, this work's method Perturbation-based QuESTT, is therefore the first unsupervised, black-box QE method for Spoken Language Translation (SLT). We evaluate our method on the largest available QE dataset for SLT, the IWSLT23 dataset by Sperber et al. [27]. It consists of human quality assessments for transcriptions, machine-generated translations and reference translations of larger audio documents in form of *TED*[1] talks and presentations given in scope of the Association for Computational Linguistics (ACL)[2]. The dataset consists of about 72.6% TED data and 27.4% ACL data. While the ACL speech contains domain-specific language and professional jargon, the TED data is more colloquial.

[1] https://www.ted.com/
[2] https://www.aclweb.org/

2.3. Perturbation

Separating from its predecessors' glass-box methodology, this work's aim is the development of an architecture-agnostic method for S2TT QE based on input perturbation. Perturbation has been a ubiquitous tool for nearly all types of Machine Learning and is commonly used on training data to increase model generalization and reduce overfitting [23], in which case it is commonly referred to as *Augmentation*. In particular, this includes the perturbation of audio data for that exact purpose, ranging from image perturbation of audio spectrograms to perturbation of the signal itself [14]. Nanni et al. not only improve task performance using audio data augmentation, but provide an extensive repertoire of various audio perturbation techniques with detailed explanations of their nature [14]. We partially draw from some of their presented perturbation strategies for Perturbation-based QuESTT.

Perturbation at test-time instead of training-time has proven to be an effective tool to achieve more robust and accurate performance [23]. Consistent predictions under perturbation are found to be an indicator of high confidence and a robust model [23]. This work builds on these insights and uses ensembled prediction on distinct perturbed variants of the same input datum to estimate model confidence and translation quality.

2.3.1. On Audio Perturbation Scope

Contrary to the discrete nature of language, its manifestation through speech results in a continuous signal of amplitude over time. Approaches to modifying this signal whilst keeping intelligibility, quality and, when intended, meaning intact present themselves in many forms.

Word-level perturbation is drawn into consideration due to it being tried and tested in Natural Language Processing (NLP), including in other QE methods [5] [11]. The alignment of continuous audio segments to source language words for word-level perturbation remains an active research task in ASR and can be performed to varying degrees of exactitude and complexity. The most accurate manner of segmentation would be the use of a speech recognition model or a neural forced aligner [1] [19].

Avoiding this problem and instead taking advantage of the continuity of the signal, **sequence-level** perturbation of the audio presents itself as a viable approach. To this end, the entire signal is modified using some function applied consistently on every sample. By itself, sequence-level perturbation has found application predominantly in data augmentation techniques for boosting training performance of ASR models, with positive results [14]. The intention behind this work's reliance on sequence-level audio perturbation stems from the thorough research which documents sequence-level audio perturbation being used to increase, as well as measure a models' robustness [14][23]. We use it to evaluate recognition invariance, prediction confidence and, consequently, speech translation quality. The assumption behind this strategy is that the indicators within human speech which give away its meaning are invariant when completely and consistently spoken faster, slower, at different pitches or in noisy environments.

3. Method Design

The essence of this work is the development of a method which, given a machine-generated speech-to-text translation and its source language audio, predicts the quality of the generated translation. As additional requirements to the integrity of the method, we stipulate that this method shall be

Unsupervised. The method's development shall not require any training data, meaning human-generated quality assessments of machine generated speech-to-text translations.

Reference-free. The method shall not use reference translations to perform its quality estimation.

Black-box. The method shall not rely on the use of any information with regard to the inner workings of the predicting system.

To construe a method which incorporates these characteristics, we choose to rely on a basis of perturbation. More concretely, this method's quality estimations shall result from a process rooted in the perturbation of the source language audio input. We can then infer translation quality based on the prediction's robustness when its input is manipulated using said perturbation.

3.1. Method Structure

Our proposed method, Perturbation-based QuESTT, is a black-box, unsupervised QE method, which given

- a source language audio and
- inference access to the predicting system

returns a quality estimate of the given translation as a number within $[0, 100]$. As illustrated in Figure 3.1, the Perturbation-based QuESTT model consists of three main components. The source language audio is perturbed using multiple preset strategies and passed to the predicting system for translation. On the basis of the resulting translations' similarity to the initially given translation, a score is calculated and returned as the quality estimate.

3.2. Perturbation

After receiving the source language audio a, the Perturbator trims leading and following *silence* from it, illustrated in Figure 3.2. *Silence* refers to a signal amplitude lower than

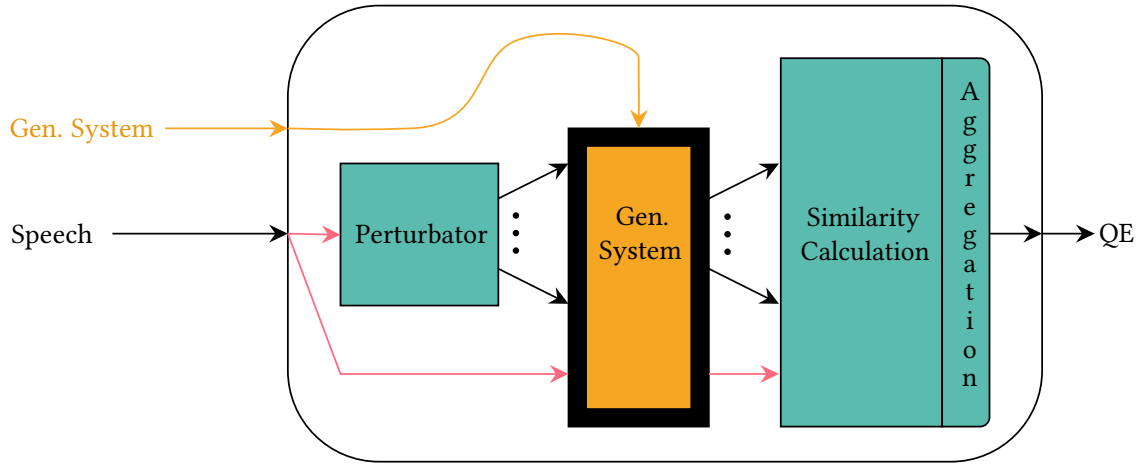
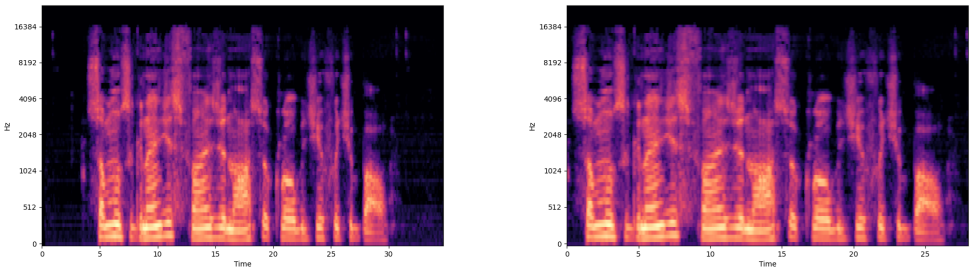


Figure 3.1.: The QE Model takes source language speech and inference access to the generating system marked in orange. The speech is perturbed and multiple variants are passed through the generating system, as well as an unperturbed reference, marked in red. Its predictions of the perturbed speech are compared to the prediction of the original given machine translation and, based on the calculated similarity, the QE-model returns its quality estimation.

a fixed threshold of decibels relative to the full scale (dBFS). The maximally possible amplitude represents 0 dBFS; lower amplitudes correspond to negative numbers up until reaching the lowest possible signal at the end of the audio's dynamic range, which is dependent on the audio's sample size. An audio with a sample size of 16-bit would have a dynamic range of 96dB and its lowest dBFS value would be -96 dBFS.



(a) Original Audio

(b) Trimmed low-amplitude sections at beginning and end with a threshold -16 dBFS.

Figure 3.2.: The Perturbator's audio preprocessing step. Leading and trailing silences are trimmed.

Using the preprocessed audio, the Perturbator creates n new, different audios $\tilde{a_1}, \ldots, \tilde{a_n}$ by applying perturbations $p_1, \ldots, p_n$ to the given audio, as shown in Figure 3.3. n and its corresponding perturbation functions $p_i(\cdot)$ are fixed parameters of the Perturbator.

Formally, the Perturbator can be written as a function

$$\forall m \in \mathbb{N}: \quad \mathcal{P}_n : \mathbb{S}^m \longrightarrow \mathbb{S}^{m_1} \times \mathbb{S}^{m_2} \times \cdots \times \mathbb{S}^{m_n}$$
$$a \longmapsto \begin{pmatrix} p_1(a) \\ \vdots \\ p_n(a) \end{pmatrix} = \begin{pmatrix} \tilde{a}_1 \\ \vdots \\ \tilde{a}_n \end{pmatrix}$$

with

$$\forall i \in 1, \ldots, n: \quad p_i : \mathbb{S}^m \longrightarrow \mathbb{S}^{m_i}$$
$$a \longmapsto p_i(a) = \tilde{a}_i$$

where m is determined by the number of given audio samples and the exact content of the sample space $\mathbb{S}$ depends on the samples themselves. For example, normalized samples at `float32` precision would yield a sample space $\mathbb{S}$ of $[-1, 1]$. The functions p_i alter parts or the entirety of the given audio a according to varying strategies, as illustrated in Section 3.2.1. The perturbations p_i may also alter the number of samples and produce audio in $\mathbb{S}^{m_i}$. To illustrate, $\mathcal{P}_n(a)$ is similar to a matrix with unevenly long rows, but always n of them.

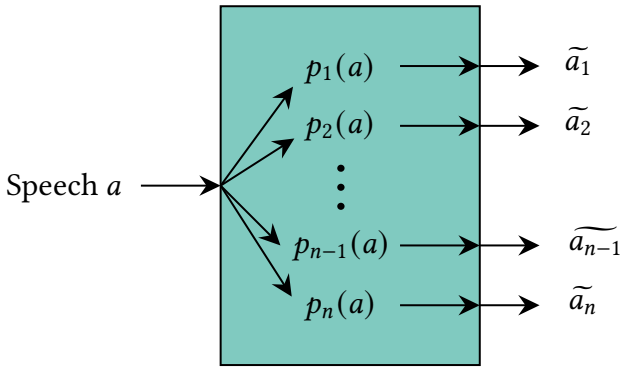


Figure 3.3.: The Perturbator (marked in green) takes a speech audio a and applies perturbations $p_1, \ldots, p_n$ to produce n perturbed audio variants $\tilde{a}_1, \ldots, \tilde{a}_n$.

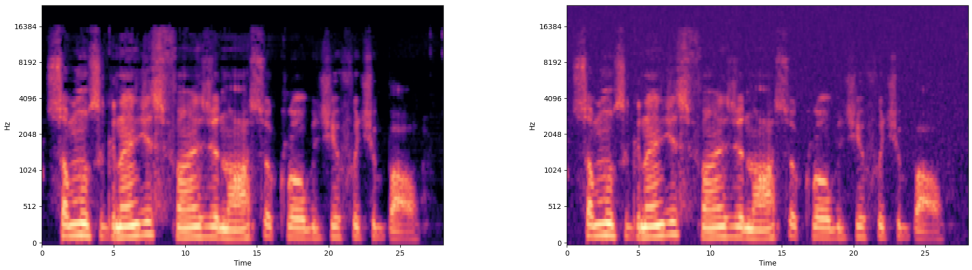
3.2.1. Perturbing Speech Signals

We perform sequence-level perturbations and decide against word-level perturbation. As to avoid easy deprecation and dependence on foreign systems in PERTURBATION-BASED QUESTT, we are neither willing to add a neural audio segmentation component to the method, nor do we wish to compromise the integrity of our *black-box*, *Speech to Text* Translation QE method by working with audio transcriptions.

To perform the sequence-level perturbations, we rely on multiple distinct strategies. Speed shifts, pitch shifts, resampling and noising are shared with Nanni et al.'s approach to audio data augmentation [14], but are produced differently in this work to better fit the test-time perturbation scenario.

3.2.1.1. Noise

We add noise to the speech audio, which is a common technique for audio data perturbation [14] [13]. Since our goal is not dataset inflation while staying close to the ground truth distribution, like in many ASR applications of audio perturbation, we see no necessity to add realistic noise, as is done by Morales et al.[13]. Instead, we add random values based on a Gaussian distribution to each audio sample. The mean of the distribution is kept at 0, but its exact standard deviation is a hyperparameter. A visualization of the noise addition is provided in Figure 3.4.



(a) Clean Audio

(b) Perturbed audio using Gaussian random noise with a standard deviation of 0.007

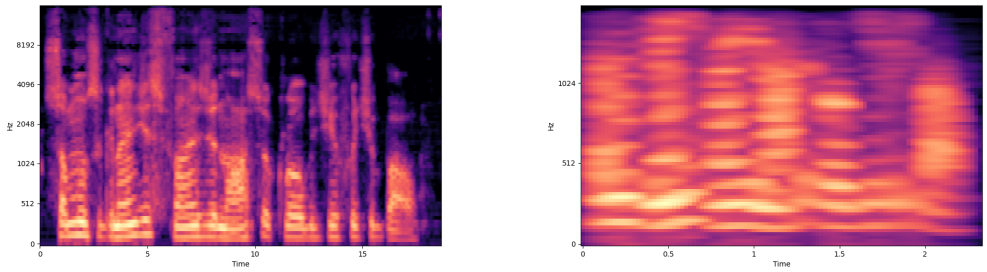
Figure 3.4.: Comparison between clean speech audio and a perturbed version of the same audio with added noise.

3.2.1.2. Resampling

To introduce variation in the technical characteristics of the audio, we utilize resampling as a perturbation method. Common resampling using Bandwidth Interpolation shows a very small amount of loss when keeping close to the original sample rate, so our choices for the sampling rate hyperparameter are restricted to extreme differences in sampling rates. When the sampling rate F_s is lowered below the Nyquist rate F_N of the audio signal, aliasing may occur [10]. The Nyquist Rate F_N is the frequency at which a signal with a bandwidth of $W = \frac{1}{2}F_N$ is still losslessly reconstruable. If an audio's sample rate is below F_N, distortion is introduced during the interpolation, which is called aliasing. We use this effect to perturb our input audio data as demonstrated in Figure 3.5.

3.2.1.3. Speed and Pitch

To mimic both changes in pitch and changes in speed, we perturb the original audio by changing its audible speed and pitch. Given the audio $a \in \mathbb{S}^m$ of duration T_s sampled at $F_s = m/T_s$, we resample a from a sample rate F_s' with $F_s' \neq F_s$ to the original sample rate F_s. The audio is sped up when $F_s' > F_s$, making it shorter as well. Conversely, for $F_s' < F_s$ the resulting audio will be slowed down and elongated.



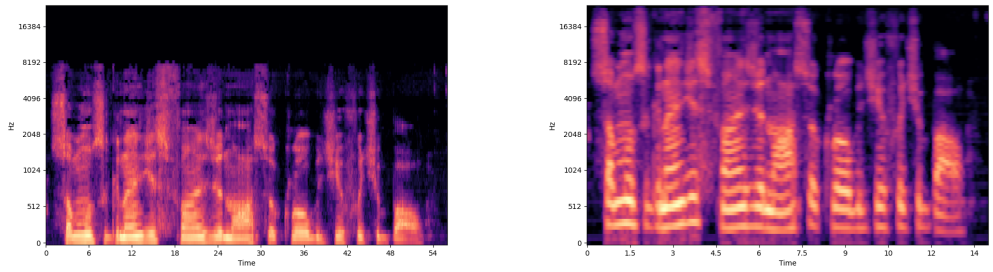
(a) Resampled to 32kHz from 48kHz. No prominent aliasing noticeable. The spectrogram of the resampled audio is almost indiscernible from the original.

(b) Resampled to 4kHz from 48kHz. Presence of aliasing clearly noticeable. Even at only 4kHz, the speech is still intelligible, although muffled.

Figure 3.5.: Perturbed versions of the original audio using resampling. The original audio (see Figure 3.2b) has a sample rate of 48kHz.

These changes in audio length for the same audio a shorten or elongate the perceived wavelengths, increasing (or lowering) the recorded waves' frequencies. The pitch of speech then sounds higher (or lower) than before the perturbation.

A visualization of these effects is provided in Figure 3.6.



(a) Perturbed audio using resampling from $F_s' = 0.5 \cdot F_s$. The duration is doubled and the frequencies are halved.

(b) Perturbed audio using resampling from $F_s' = 2 \cdot F_s$. The duration is halved and the frequencies are doubled.

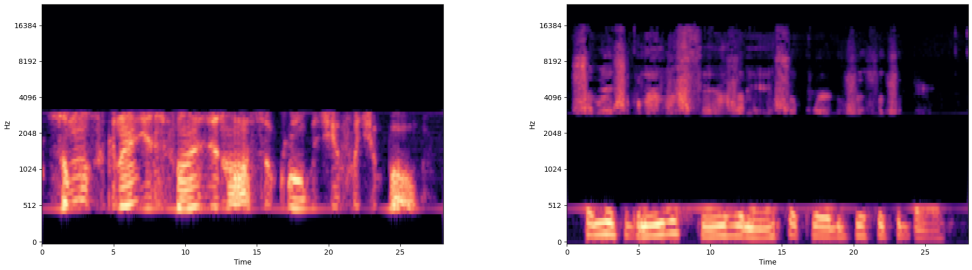
Figure 3.6.: Speed- and pitch-warping perturbation of the original audio (see Figure 3.2b. Changes in frequency and duration can be observed.

3.2.1.4. Filtering

If an SLT-models' predictions are highly dependent on frequencies which are not usually employed for speech, it might be an indicator of a bad translation. To differentiate which parts of the frequency spectrum the evaluated predictor pays attention to, we employ band-pass and band-stop filtering on the speech audio. The lower and upper bounds of the filters are hyperparameters.

To apply the required filter, a Fast Fourier Transform (FFT) [3] is applied to the given

source audio $a \in \mathbb{S}^m$ to get its discrete Fourier Transform (DFT) $\breve{a} = DFT(a)$. Then, either the frequency bins containing frequencies between the lower and upper bounds are zeroed out (band-stop), or everything outside those bounds is (band-pass). To project $\breve{a}$ back into $\mathbb{S}$, the inverse DFT is calculated using the inverse FFT algorithm. The resulting audio is the filtered perturbed version, as seen in Figure 3.7. Since the Fourier Transformation is not lossless, inaccuracies may taint the exactness of the filter cutoff and make for "smudged" filter edges.



(a) Perturbed audio using a band-pass filter with bounds $(500Hz, 3000Hz)$.

(b) Perturbed audio using a band-stop filter with bounds $(500Hz, 3000Hz)$.

Figure 3.7.: The preprocessed audio, as seen in Figure 3.2b is filtered using band-pass and band-stop filters with the same bounds. Unclean edges are the results of inaccuracies during FFT.

3.3. Translation Comparison

The QE-Head of Perturbation-based QuESTT calculates the perceived deviation of the perturbed translations from the initially given machine translation. Thus, it receives the initially given machine translation t, as well as n predicted textual translations $\tilde{t}_1, \ldots, \tilde{t}_n$ of each of the perturbed audio samples $\tilde{a}_1, \ldots, \tilde{a}_n$. We present two variants for the translation comparison: pairwise and corpus-like.

Pairwise translation comparison calculates the deviation of each $\tilde{t}_i$ from the initially given t using BLEU [16], the inverse Translation Error Rate (TER) $(1 - TER)$ [25] or the CHRF-score [18], all scaled to fall within $[0, 100]$. A function $\Delta(\cdot, \cdot)$ shall be used in place of the interchangeable translation distance metric.
The calculated deviations $\Delta(\tilde{t}_i, t)$ are then aggregated using a weighted sum with fixed weights $w_1, \ldots, w_n \in \mathbb{R}$ to make the cumulative change CC:

$$CC = \sum_{i=1}^{n} w_i \cdot \Delta(\tilde{t}_i, t)$$

To ensure consistent and comparable outputs, CC is normalized to fall within $[0, 100]$ and becomes the QE-model's quality estimation QE.

$$QE = \frac{CC}{100 \cdot \sum_{i=1}^{n} w_i}$$

This process of aggregation and normalization for pairwise translation comparison is illustrated in Figure 3.8.

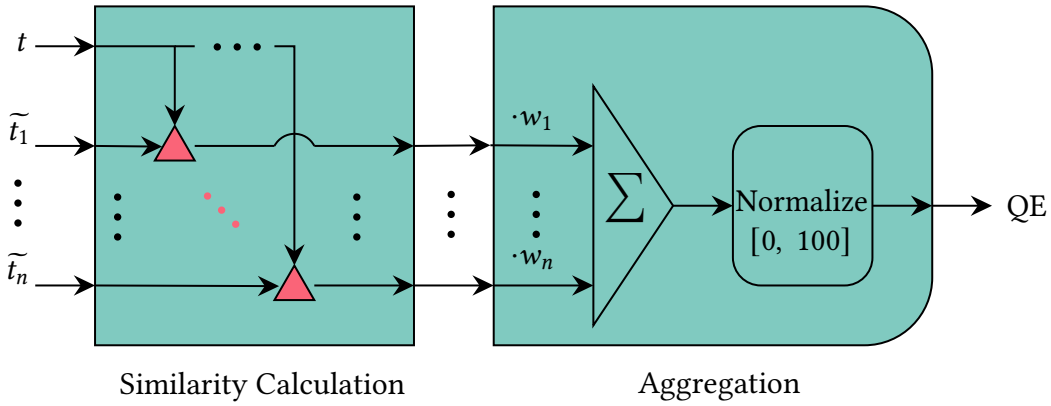


Figure 3.8.: The QE-Head (marked in green) receives the predicted translations of the perturbed audio samples $\tilde{t}_1, \ldots, \tilde{t}_n$, along with the initial machine translation t. For each $\tilde{t}_i$, the translation similarity between $\tilde{t}_i$ and t is computed using a predefined translation or sentence similarity metric (marked in red). The weighted sum of these similarities using predefined weights $w_1, \ldots, w_n$ is normalized to a score within $[0, 100]$ and returned as the quality estimate.

Corpus-like translation comparison utilizes the intended similarity of the translations of the perturbed audio $\tilde{t}_i$ and the initially given predicted translation. The set $\{\tilde{t}_1, \ldots, \tilde{t}_n\}$ is interpreted as a corpus and passed as such to a corpus-based metric Δ like BLEU, which then calculates the similarity of t to its perturbed counterparts. The resulting score scaled to $[0, 100]$ makes for the QE-model's quality estimation

$$QE = \Delta(t, \{\tilde{t}_1, \ldots, \tilde{t}_n\})$$

The corpus-like approach makes for easier hyperparameter setting, as there are no weights to set for each perturbed version of the source audio the $\tilde{t}_i$ were predicted for. However, the ability to balance out more grave perturbations and weaken the stabilizing influence of less altering ones forces an alignment of perturbation severity, possibly undercutting variance.

3.4. Weaknesses

The perturbation-based approach comes with inherent drawbacks. Procuring a quality estimation for a single audio-translation pair takes an additional n inference passes, multi-

plying the time and effort it took to translate in the first place. It is therefore difficult to optimize time complexity considering this conceptual drawback.

The pairwise translation additionally fails to exploit mutual information and requires a metric call for each $\tilde{t}_i$ instead of one for all. This adds to computational complexity. However, corpus-like translation comparison does not include a weighing process for the different perturbation strategies. Making some perturbations more influential than others is therefore not possible following that strategy.

Additionally, the choice of perturbation strategies may prove to be suboptimal. Out of the many possible audio perturbation methods, only a few of the available perturbations are selected [14]. To improve performance of quality without overextending time complexity, we attempt to collect a representative assortment of perturbation strategies. The selection for Perturbation-based QuESTT may not offer a sufficient amount of variance in the type of obscured information. Especially speed-warping, which is essentially simultaneous pitch- and speed-shifts, may prove to be more useful separated into the two components than it is as a combined variant.

4. Experimental Setup

Considering the use cases for Perturbation-based QuESTT, we formulate qualitative goals regarding the method we seek to achieve. We aim to keep the method as agnostic as possible. On the one hand, this refers to the type of translating system and the fact that this type should not signify. On usage of the quality estimation method, the only information known about the system is that it takes speech input in a source language, and is able to generate a textual translation of the source audio into a target language. The method should be functional for every such system without additional assumptions or requirements.
On the other hand, we wish to avoid the curse of easy deprecation. The conscious decision against the integration of neural components or resources on the verge of deprecation as essential method elements follows the wish for method continuity, especially considering the high speed at which the field of Machine Learning and the tools used therein have been changing in recent times.

During the experimental setup, the most important values we considered were agnosticity and independence, as per our qualitative method goals, thus complete optimality in terms of computation and time was not prioritized. However, some optimizations were made, especially for the inference passes through the evaluated predictor system, by distributing calculations between GPU and CPU.

Our QE model implementation is written predominantly using `python`[1] to be compatible with modern ML resources, but also because it is lightweight in its syntax, very open and adaptable, which aligns well with our qualitative goals. Python package management enabled us to track the implementation's requirements, while customizing to operating system requirements and availability of a GPU. This makes reproduction of this work's results easier and more straightforward.

On a higher level, our architecture mimics the core components of Perturbation-based QuESTT as illustrated in Figure 3.1 in the Method Design Section 3. Each component is represented as a separate class and can be instantiated on demand, where the `Quality_Estimator` class controls the flow of information and acts as the access point for inference and evaluation.

To provide maximal flexibility for hyperparameter customization, the evaluation and inference scripts for our QE model may be configured by `.json` configuration files containing a list of perturbation configurations, their relative importance, the translation similarity metric and whether it should be passed the predicted translations of the perturbed audios as a corpus or individually. For evaluation purposes, multiple distinct such configurations

[1] `https://www.python.org/`

are bundled and evaluated on the bwUniCluster (v2)[2]. We, the authors, acknowledge support by the state of Baden-Württemberg through bwHPC.

For instructional information concerning result reproduction and work environment setup, refer to the `README` in our provided GitHub Repository[3].

4.1. Data Preprocessing

We use the IWSLT23 dataset for evaluation. It contains audio segments mapped to a reference transcription, translation and a machine generated translation with a human annotator score. These segments are, however, not fully coordinated with the corresponding segmented audio files, making data preprocessing necessary to adapt it to our scenario. The corresponding audio data to the annotated translations in the dataset was acquired from IWSLT [4]. The TED audio was provided in shape of entire talks with durations partly longer than ten minutes. Therefore, it was segmented using an accompanying YAML file containing timestamps for each audio file and text segment spoken in it. Some resulting audio segments were removed due to faulty timestamps producing audios that contained no speech or were too short for inference. As the number of resulting audio segments is very large, not all were checked for correct- and completeness.

4.2. Resources

4.2.1. Handling Models

Any interaction with the translation generating system passes through a model wrapper, which provides inference access and can be configured with the correct model information upon instantiation of the class. Keeping inference access to the evaluated translation systems as uniform as possible through this model wrapper follows our qualitative goals for translation system agnosticity.

By working with widely-used libraries like *huggingface*'s `transformers` library[5] and `torch`[6], we aimed to make integrating and evaluating own models as easy as possible. For demonstration and method evaluation purposes, some models have been previously embedded. In addition to this, further models can be loaded from huggingface by adding a model to the supported huggingface key list, with a custom inference method if necessary. Custom models can be loaded from a local path by providing a model loading method using a corresponding implementation of `torch.nn.Module` and a fitting inference method within the model wrapper.

When embedding *Meta*'s *Seamless* model from huggingface, we found discrepancies between a manually calculated output sequence probability and the specially provided

[2] `https://wiki.bwhpc.de/e/BwUniCluster2.0`
[3] `https://github.com/13thWitch/QE-for-S2TT`
[4] `https://iwslt.org/2023/multilingual`
[5] `https://huggingface.co/docs/transformers/en/index`
[6] `https://pytorch.org/docs/stable/torch.html`

huggingface method for that purpose[7]. As the manually calculated output sequence probabilities were unusually tiny, we use the *huggingface* method in our evaluation.

4.2.2. Handling Audio

We load audio using the lightweight `soundfile`[8] module, which provides integrated support for `.mp3` as well as `.wav` files. Working with the resulting `numpy` array proves to be convenient and allows for transparent audio editing. For example, using the renowned `pydub`[9] module, we are able to detect leading and trailing silences and trim them from given audio.

Initially, we had planned on using a popular audio augmentation library named *SpecAugment* which implements time and frequency warping on mel spectrograms [17]. Unfortunately, the perturbations performed using SpecAugment are too insignificant to force any perceivable change to human hearing, let alone in machine predicted translations. Although this could have been mitigated by making some manual adjustments, additional version compatibility issues and performance issues could not be overlooked. The conversion of the source language audio to mel spectrogram and back either came with an audible loss in audio quality, or at the cost of long inference times of an embedded neural Vocoder system. The combination of SpecAugment and our chosen state-of-the-art Vocoder, HifiGAN [9], was therefore not feasible. We then favoured a manual implementation of the audio perturbation methods instead, following our quality and independence goals and avoiding easy deprecation.

4.3. Implementing Perturbation

The `Perturbator` class is configured upon instantiation using the passed desired perturbations. When it is called upon to generate perturbations for a given audio, this audio is trimmed before it is passed on to the respective perturbation strategy functions with the requested specifications. For the remainder of this section we will be focusing on the implementation of these individual perturbation methods.

4.3.1. Resampling and Speed-Pitch-Warping

The resampling process was performed using the python `resampy` [10] module, which implements the band-limited sinc interpolation method for sampling rate conversion following Smith's work on Bandlimited Interpolation and the corresponding algorithm presented therein [24]. On certain audios `resampy` outperforms many similar tools in terms of speed by significant margins. We resample as a perturbation method to frequencies well below

[7] `https://discuss.huggingface.co/t/announcement-generation-get-probabilities-for-generated-output/30075`
[8] `https://pypi.org/project/soundfile/`
[9] `https://github.com/jiaaro/pydub`
[10] `https://github.com/bmcfee/resampy`

(e.g. 8kHz) and well above (e.g. 32kHz) the source language audio's Nyquist Rate (e.g. 16kHz in the IWSLT23 dataset).

To mimic variations in speech velocity, we resampled the audio to its original sample rate while assuming it was sampled at a different sample rate using `resampy`. The thereby slowed or accelerated speech changes in pitch as well, testing for pitch-agnostic phoneme recognition at the same time. After hyperparameter tuning, speed perturbations within $[0.5, 2]$ times the original sample rate proved most effective for the QE task.

4.3.2. Frequency Filtering

To restrict the frequency spectrum of the source language audio, we use band pass and band stop filters. They either restrict to

- a certain section of the human speech base frequency spectrum,
- its formant-giving harmonics, or
- to outside of it, letting only noise at frequencies outside the human speech spectrum pass.

We apply these filters using the python `scipy.ftt` module [11]. We reconstruct the audio to shift the data's x-axis from indicating time to indicating frequency and apply a simple binary mask on top, depending on the filters' cutoff. We reconstruct back to the time-based view with the inverse fourier transform available in `scipy.fft`.

4.3.3. Noising

For most of our computations regarding audio, we work with `numpy` [12]. This includes the addition of Gaussian Noise as a perturbation strategy. To this end, we use `numpy` to generate an array of random numbers drawn from a normal distribution with a mean of 0 and an experimentally determined standard deviation within $[0.001, 1.0]$.

[11] `https://docs.scipy.org/doc/scipy/tutorial/fft.html`
[12] `https://numpy.org/`

5. Results

We evaluate using only *Meta*'s *Seamless M4T v2* model, a state-of-the-art multi-modal translation system [4], which we apply as a cascade Speech-to-Text Translation System. As evaluation data, we use the first 90 entries of the IWSLT23 dataset [27]. Out of the three available language variants in the IWSLT23 set, we work with the `en-de` split to tune parameters and hyperparameters. Using reference-based COMET [20], we compare the translation quality predicted by our QE model to the COMET score given the predicted translation and the references for translation and transcription provided in the IWSLT23 data. Iteratively, we perform loosely structured parameter and hyperparameter tuning. The decision against systematic approaches like grid-search was made due to a lack of computational resources and time for extensive hyperparameter optimization. A more extensive account on our evaluation's execution times can be found in Section 5.2.

We present multiple variants of our QE model, using each of the three translation comparison metrics in pairwise and corpus-like fashion. The translation similarity calculation strategy and metric are treated as hyperparameters and define our model variants. On their basis, we tune our remaining parameters, which includes exactly which perturbations are performed and, if performing pairwise translation similarity calculation, weights for each applied perturbation.

As indicated in Table 5.1, our best-performing variant by Pearson Correlation with the reference-based COMET scores is PB-QUESTT-chrf*, configured with corpus-like translation similarity calculation using CHRF. This model variant solely employs resampling at target sample rates 21kHz, 22kHz, 23kHz, 24kHz and 25kHz as perturbation strategies. Following the MAE and the RMSE to the reference-based COMET scores, PB-QUESTT-chrf performs best. It employs

- added Gaussian noise at standard deviations 0.1, 0.5, and 0.7,
- warping at factors 0.66 and 2.0, and
- a frequency band pass filter with bounds $[100, 3000]$

as perturbation strategies, performing a total of six perturbations.

The configurations of the other mentioned models in Table 5.1, as well as those of the performed ablations and single-perturbation tests shown in Table 5.2, can be found in the corresponding Section A.2 in the appendix.

5.1. Robustness

Additionally to the quantitative performance evaluation we complete, we evaluate our QE Model's performance in various input and parameter settings to test its quality and robust-

QE variant	Pearson	MAE	RMSE
PB-QuESTT-chrf*	**0.598579465070**	12.00920995	15.6684637964
PB-QuESTT-chrf	0.519477215283	**11.87913072**	**14.5200751896**
PB-QuESTT-bleu*	0.518029160540	17.51945238	22.1468098974
PB-QuESTT-bleu	0.512772074737	17.30039246	21.8373347467
PB-QuESTT-ter*	0.484574537952	17.53206198	22.7090405640
PB-QuESTT-ter	0.217075104305	43.10477027	46.7448233976
Sequence Prob.	0.120589519967	79.41471707	78.0789525817

Table 5.1.: Performance of our QE model variants as relation to COMET scores on *Seamless* translations. The similarity between our scores and COMET scores is given as rounded Pearson correlation, the Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE). An asterisk (*) next to our model variant designation marks corpuslike translation similarity calculation. We provide the sequence probability of *Seamless* as a baseline. The bold scores are the best for the respective metric.

ness. We perform ablations of PB-QuESTT-chrf, our best-performing model according to two out of three evaluative metrics. We take a closer look at the effect each perturbation strategy has on model predictions, and test QE model performance on out-of-distribution inputs.

5.1.1. Ablation Studies

Aiming to more closely inspect each perturbation strategy's contribution to Perturbation-based QuESTT, we evaluate QE model variants configured to only perform one type of perturbation. Each of these models can, however, use different perturbation specifications, for example multiple noising perturbations varying only in the standard deviation of their random distribution and the respective weights.
We provide a frame of reference for the ablated perturbations by additionally comparing to the best-performing parameter and hyperparameter configuration we found which uses only one kind of perturbation each. These cumulative results, as listed in Table 5.2, show each ablation performing significantly worse than its counterpart. It follows, that individual good performance of a perturbation strategy is not a direct indicator of good performance in combination with other perturbation strategies, and vice versa.

5.1.1.1. Pure Resampling

When perturbing the source language audio only by resampling it to sampling rates well below the audio's Nyquist Rate, we observe quite strong results, demonstrating a Pearson correlation with the reference COMET [20] predictions of up to ≈ 0.5986. We find that for source audios with a sample rate of 48kHz, pure resampling to sample rates within $[21000, 25000]$ Hz proves to be the most effective specification we tested.
In general, the amount of change in the observed model's translation resulting from

QE variant	Pearson	MAE	RMSE
Filter only	0.3782313554	19.758169403	26.051006034
Filter best	0.3336799863	13.938110087	19.595256695
Noise only	-0.0561381232	49.187997378	51.857298658
Noise best	0.4906354995	19.461261940	24.439950805
Warp only	0.0627406393	33.388123066	37.676180935
Warp best	0.4124354996	14.970776571	19.996201139
Resample best	0.5985794651	12.009209946	15.668463796

Table 5.2.: Similarity of ablations and single-perturbation QE model variants to COMET scores given as rounded Pearson Correlation, Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). Note that *Resample-optimal* is equivalent to PB-QuESTT-chrf*.

the perturbation is severe and observable. The increasing degradation of the produced translations when the perturbed audio's sample rate falls below the original audio's Nyquist Rate is apparent and exemplified in Figure 5.3. We therefore conclude that the resampling target interval of $[21000, 25000]$ Hz proves to be the tipping point for intelligibility of speech audio.

Sample Rate [Hz]	Translation
26000	We are big fans of our football club.
25000	We are big fans of our football club.
24000	We're the biggest fans of our club.
23000	"We are great fans of our club, the Football Club."
22000	Five thousand dollars for you and your wife.
21000	What's the matter with you?
20000	I'll go with you.
19000	I'm going to take a look at it.
18000	I'm going to do it.
17000	I'm not going to do it.
16000	I'm going to go back to the beginning.

Table 5.3.: Note the translation's deterioration with lowered sample rates. The original audio contains Portuguese speech which translates to "We are big fans of our football club.". The original audio is sampled at 48kHz and has a bandwidth of $\approx$ 15kHz. The translations are produced by *Seamless*.

5.1.1.2. Pure Noising

A mere addition of noise to the source language audio proves to be a relatively good perturbation strategy, correlating with the reference-based metric COMET [20] at more than 0.49 Pearson. Perturbing example audio reveals that a robust translation can stay

stable, even using perturbation with added Gaussian noise with large deviations like 0.2 or higher, as can be seen in Table 5.4. However, the best performing QE-Model variant only using added noise as a perturbation strategy employs Gaussian noise with a comparably very small standard deviation within $[0.003, 0.012]$. We interpret that, although some translations stay stable at high standard deviations, stabilization and low-impact perturbation better suit the QE task. A reason for this discrepancy may be that *Seamless* demonstrates state-of-the-art quality and a bias toward positive quality estimations when evaluating on its translations might raise performance.

Standard Deviation	Translation
0.1	We are big fans of our football club.
0.2	We are big fans of our football team.
0.3	We are big fans of our team.
0.4	We're in the middle of our first day of the festival.
0.5	for our big fans.
0.6	We're not discussing anything about this.
0.7	I'm going to take a look at the video.
0.8	I'm going to go to the bathroom.
0.9	It's not like I'm going to be able to do it.

Table 5.4.: Note the translation's deterioration with rising standard deviations on the added Gaussian Noise. The original audio contains Portuguese speech which translates to "We are big fans of our football club.". The translations are produced by *Seamless*.

5.1.1.3. Pure Warping

We use warping to emulate varying pitches and velocities of speech. To extract the factor thresholds at which the generated translations react to the perturbation, we perform an exemplary analysis of the transition of the *Seamless* predicted translations from consistent prediction of the original translation to increasingly deviating and nonsensical generated sequences, as depicted in Table 3.6. It identifies values around 0.66 and 2.1 as the warp factors at the predictions' tipping point, with values closer to 1.0 naturally increasing in similarity to the originally predicted translation.

However, the ablation from the PB-QuESTT-chrf model only performing warping at the tipping point factors 0.66 and 2.0 performs catastrophically, reaching a Pearson correlation of near 0 with the reference-based COMET scores, as presented in Table 5.2. The optimal warping configuration instead applies warping to a range of factors, starting from the tipping points and extending inward towards the original at factor 1.0. This approach performs more than four times better than the two tipping point values on their own, further supporting the notion that slightly more conservative perturbation over finer grained parameter intervals proves more effective.

Warp Factor	Translation
0.5	The first thing that comes to mind is the fact that the world is changing.
0.55	The nine verses of the Qur'an are the verses of the Qur'an.
0.6	The nine-year-old is not a child, he is a child, he is a child .
0.63	The first thing I want to do is to make sure that the people who are going to be in the room are safe.
0.66	They're our biggest fans, our football club.
0.7	We are big fans of our football club.
⋮	⋮
1.8	We are big fans of our football club.
1.9	"We are big fans of our football club"
2.0	We're the biggest fans of our club.
2.1	"We are the great brothers of our football club."
2.2	"We're going to the festival."
2.3	I'm going to take a shower.

Table 5.5.: Note the translation's deterioration increasing with warp factors further away from the original at 1.0. The original audio contains Portuguese speech which translates to "We are big fans of our football club.". The translations are produced by *Seamless*.

5.1.1.4. Pure Filtering

To extract which part of the frequency spectrum of the source language audio the predicting model pays attention to when generating its translation, we apply frequency band filters as a perturbation. Our results only using this perturbation method prove to be quite mediocre compared to the other presented perturbations and ablations in this work. This fact is expected given the comparatively higher parameter complexity in frequency band filtering. Not only is every filtering-based perturbation characterized by whether it is a band pass or band stop filter, each bound is another free parameter to be set. Our unautomated process of parameter optimization presumably fails to find completely, or even approximatingly, optimal filter settings for speech perturbation, therefore generating poorer results than the other perturbations.

However, we do gain some insight from inspecting the variation of predicted translations on a variously filtered speech audio example, the results of which can be inspected using Table 5.6. The frequency range of [50, 600] Hz seems to contain the core speech components. Multiple layers of harmonics are stacked on top of these frequencies, so that a sufficiently wide band containing these harmonics, e.g. [1000, 7000] Hz can reproduce the missing information to a certain extent. Some perceived threshold bounding values therefore seem to be a band of at least 500Hz width within the scope of the core speech components, and filters which rely only on harmonics with a bandwidth of about 6000 Hz. Further examination of the relevant frequency bands is required to improve the effectiveness of filtering as a perturbation strategy.

Type	Bounds	Translation
PASS	(100, 1100)	We are the big fans of our football club.
PASS	(50, 1000)	We are big fans of our football club.
PASS	(1000, 7000)	We are the key to the foundation of our football club.
PASS	(500, 3000)	What's the matter with you? What's the matter with you? What's the matter with you?
STOP	(100, 1000)	We are big fans of our football club.
STOP	(2000, 5000)	We are big fans of our football club.
STOP	(200, 1200)	We are the big ones in the soccer club.
STOP	(550, 4500)	I'm a big fan of the football club.
STOP	(500, 5000)	We are the great fans of the team that I got from you.
STOP	(450, 5500)	We're great friends, and we're all good friends.

Table 5.6.: Translation variance under filtering perturbation. The original audio contains Portuguese speech which translates to "We are big fans of our football club.". The translations are produced by *Seamless*. The bounds are given in Hz.

5.1.2. Out-of-distribution Performance

We performed our quantitative evaluation on *SEAMLESS*, a state-of-the-art S2TT-model. To evaluate the behaviour of our QE model on out of distribution data, meaning an atypical

S2TT model predicting atypical translations, we implemented a trivial predictor which always outputs the same German sentence *"Dieser Satz is eine sehr eintönige Übersetzung, finde ich."*. This trivial predictor is therefore maximally robust, as its predictions do not deviate under perturbation, but demonstrates catastrophic performance, as it is almost never correct.

We assume that our perturbation-based QE method which infers quality through robustness would give overwhelmingly and erroneously positive estimations for the predicted translations. This assumption is confirmed after an evaluation of the trivial predictor on a subset of the IWSLT23 dataset, indeed revealing maximal predicted quality scores by our QE Model and a Pearson correlation with the reference-based COMET scores of $\approx -8.13 \cdot 10^{-16}$.

5.2. Runtime Evaluation

The applicability of our method to real-time translation scenarios depends on low inference times of our quality estimation model. We analyze execution time and QE-model inference time during our evaluation to examine the implementation's applicability. The runtimes are given for evaluation on an Intel Xeon Gold 6230 GPU with four NVIDIA Tesla V100 accelerators[1] unless otherwise indicated.

We collect inference and evaluation execution times in Table 5.7 and record inference times of at least 3.7 seconds. This inference time is tolerable for asynchronous use, but for real-time systems, potentially as an embedded system with further computation based on its result, this implementation of Perturbation-based QuESTT does not seem applicable.

When comparing identical perturbation configurations and their runtimes from Table 5.7, once with pairwise and once with corpus-like translation similarity calculation, which is the case for PB-QuESTT-bleu and PB-QuESTT-bleu*, we observe that, contrary to our intuition, one larger pass through the metric instead of many smaller ones is not faster. In fact, as can be extracted from Table 5.8, in two out of three cases, the inference times for the corpus-like computation with the same metric proves slower than the pairwise translation similarity calculation.

However, the outliers in Table 5.8, PB-QuESTT-ter and PB-QuESTT-ter*, not only differ in the number of performed perturbations, but also in the kinds of perturbation used. While PB-QuESTT-ter* uses noising, warping and filtering, its counterpart PB-QuESTT-ter uses noising, warping and resampling. To investigate the runtime differences between different kinds of perturbations, we compare the runtimes of single-perturbation-method QE-model variants in Table 5.9. They reveal that this outlier can either be explained by the higher time cost of using resampling as a perturbation method over filtering. We additionally observe that added Gaussian noise proves to be the fastest perturbation strategy.

However, we only use these observations as indicators and do not postulate absolute fact. We notice that these differences in runtime do not remain completely consistent over

[1]Exact device specifications can be found under GPU_4 on the bwUniCluster `https://wiki.bwhpc.de/e/BwUniCluster2.0/Hardware_and_Architecture`

QE Model	Median I.T.	Mean I.T.	Execution Time
PB-QuESTT-chrf	4.911639	5.5412335	738.3733
PB-QuESTT-chrf*	4.633770	5.0436238	698.8798
PB-QuESTT-bleu	3.799485	4.1151464	617.4134
PB-QuESTT-bleu*	3.890306	4.2727717	674.3913
PB-QuESTT-ter	6.796510	7.5684684	906.0613
PB-QuESTT-ter*	4.347694	4.8486077	664.2408

Table 5.7.: Median and Mean Inference Times (I.T.) and evaluation execution time for our QE model variants, in seconds.

QE Model	Normalized Mean Inference Time
PB-QuESTT-chrf	0,923539
PB-QuESTT-chrf*	1.008725
PB-QuESTT-bleu	1.028787
PB-QuESTT-bleu*	1.068193
PB-QuESTT-ter	0.840941
PB-QuESTT-ter*	0.808101

Table 5.8.: Mean Inference Times (I.T.) for our QE-Model variants, in seconds, divided by their corresponding number of performed perturbations. Example: PB-QuESTT-chrf performs 6 perturbations: 3 noising, 2 warping, 1 filtering.

the same evaluation data. Therefore, to make definitive statements on the exact runtime differences, a larger evaluation set is needed.

QE Model	Normalized Median Inference Time
Filter best	0,883220
Noise best	0,821450
Resampling best	0.863926
Warp best	0.834579

Table 5.9.: Median Inference Times of single-perturbation-method QE-variants evaluated on a subset of our evaluation data. It is given in seconds, divided by the number of performed perturbations. All configurations of the listed variants specify corpuslike CHRF. The runtimes were collected on the GPU_8 of the bwUniCluster.

5.3. Discussion

Our results confirm that Perturbation-based QuESTT is indeed a viable method for SLT quality estimation. The measured high Pearson correlation with reference-based COMET values surpasses our output sequence probability baseline by almost five times its value. The RMSE and MAE to the COMET scores of our best-performing QE-model-variants, PB-QuESTT-chrf and PB-QuESTT-chrf*, are lower than 16%. We achieve our qualitative goals, including retaining model agnosticity and having avoided the embedding of easily deprecated components.

However, the method demonstrates inherent flaws on out-of-distribution data, which can cause dramatic drops in performance, for example when a bad model is very robust. Additionally, our recorded runtimes, while acceptable for asynchronous QE-model inference, prove too slow for use in a real-time scenario. We are also not able to make any definitive assessments of which translation similarity calculation strategy is faster, as the margins by which they deviate are too small and inconsistent. As our evaluation set is quite small due to the very limited computational resources at our disposal, the reliability of our results, regarding performance and runtime, can not be unequivocally guaranteed. A larger-scale evaluation is needed to confirm the insights we draw from our limited experiments.

The perturbation framework used in this work fulfills its purpose completely. Using its perturbations, we register added Gaussian noise as the fastest perturbation, and reach peak performance of our QE-model implementation using only resampling as a perturbation technique. In general, more conservative and less aggressive perturbation specifications prove to be more effective for quality estimations.

6. Conclusion

Quality Estimation for Spoken Language Translation has barely received any attention from the scientific community. To our knowledge, of the methods available none are applicable to a fully black-box scenario and none are unsupervised. This work fills that gap by presenting the novel method Perturbation-based QuESTT. Not only is it unsupervised and requires no training or training data, it also exclusively relies on black-box information. This makes it applicable to a wider array of usage scenarios where system-specific information may be inaccessible, for example because the predicting translation system is proprietary and its API too restricted. Additionally, the lack of training, especially on domain-specific data, makes Perturbation-based QuESTT applicable to all domains and predicting translation systems.

Drawing from previous work on audio data augmentation, we integrate simple and fast audio perturbation into our method. The translation system's resulting predictions for perturbed audio are then used as indicators of translation robustness and confidence. We connect these qualities to overall translation quality, characterizing this work's core assumption. This assumption holds merit, as we discover during our experimental evaluation, measuring a Pearson Correlation to the reference-based COMET scores of more than 0.59, almost five times that of the output sequence probability baseline. We also record a mean absolute error to the COMET scores lower than 12%. With relatively sparse parameter and hyperparameter tuning, we achieve good QE performance using multiple different variations of our translation similarity calculation between the predictions on perturbed and unperturbed audio.

Next to our standard performance evaluation, we analyze runtime 5.2, out-of-distribution performance 5.1.2 and ablations 5.1.1. Giving concrete examples, we find the parameter intervals for which the presented perturbation strategies prove effective, changing the speech audio enough to prompt a response in the predicted translation, but not so much as to make it unintelligible. We find that resampling proves to be the most effective perturbation strategy for estimating quality, but also uncover potential in other perturbations, which may be realized after more hyperparameter tuning. Due to limited computational resources, we only perform manual hyperparameter tuning and only evaluate on a small dataset, which calls into question the unequivocal reliability of our results.

Additionally, we test for our method's weaknesses. The method's foundational assumption, that robustness can indicate quality, falls short when predictors are very robust and confident in their translations, but rarely correct. We explicitly showcase a staggering performance drop caused by an extreme instance of this phenomenon. However, even at peak performance, our method exhibits flaws concerning its applicability. The concept of a perturbation-based method requires repeated inference of the same translation system, making each inference pass through the QE model potentially many times longer than the initial translation's generation. Especially in real-time systems, which are quite common

in SLT, an increase in duration of this magnitude could render the method unusable in certain scenarios.

Nevertheless, Perturbation-based QuESTT proves to be an effective method to estimate the quality of most SLT models' translations, its applicability depending on the use case. Our experimental evaluation of it also yielded a fully-functional implementation of an independently usable audio perturbation framework for frequency band filtering, noising, resampling and speed-pitch-warping, providing foundational resources for future work using audio perturbation.

6.1. Future Work

Aside from the conceptional weaknesses iterated above, there is much potential to be found for improvements upon this method.

Resulting from the strict time-constraints of a thesis and inexperience with the resources at hand, we perform quite sparse (hyper-)parameter tuning in this work. It is therefore quite probable that the optimal parameters and hyperparameters have not yet been discovered and that the maximum performance yield of our method has not yet been reached. Further (hyper-)parameter tuning strategies like grid search being applied to polish this method could vastly increase its performance. The necessity for polish also applies to our evaluation: Evaluation on such a small data subset as ours may not yield fully reliable results. An additional, more extensive evaluation in terms of evaluation data and diversity in the translating model would improve credibility and usability of the achieved results. To ensure domain independence, additional evaluation on the French-English QE dataset by Besacier et al. [2] could be performed.

As the implementation provided in this work was not built to optimize performance in terms of temporal and computational complexity, a grid search using this implementation may be quite expensive. An improved implementation, focused on maximal parallelization and device optimization, would not only ease the process of parameter and hyperparameter optimization, but conversely lower the implemented QE model's inference time, making the method's application more practically feasible.

On a larger scale, we see potential in combining more of the audio data augmentation strategies presented by Nanni et al. [14] with our perturbation-based approach to quality estimation for SLT. They not only perform perturbations on the audio signal, but also on the audio's spectrograms, drawing from computer vision data augmentation strategies as well. We find this promising considering the widespread use and extensive research of image data augmentation in Computer Vision [23].

Finally, we recognize the merit of a word-level QE method for SLT in line with Dinh and Niehues' perturbation-based approach to QE for text-to-text translation [5]. Using a sufficiently performant word boundary identification system for speech audio, the perturbation framework built in the scope of this work could be used to perform word-level audio perturbations by only perturbing certain audio segments within the given speech. We provide an example of this possibility through the implementation of segment-level perturbation into our audio perturbation framework. Examples of this segment-level perturbation can be inspected in the appendix A.1.

Bibliography

[1] Robin Algayres et al. *XLS-R fine-tuning on noisy word boundaries for unsupervised speech segmentation into words*. arXiv:2310.05235 [cs, eess]. Oct. 2023. DOI: 10.48550/arXiv.2310.05235. URL: http://arxiv.org/abs/2310.05235 (visited on 09/09/2024).

[2] L. Besacier et al. "Word confidence estimation for speech translation". In: *Proceedings of the 11th International Workshop on Spoken Language Translation: Papers*. Ed. by Marcello Federico, Sebastian Stüker, and François Yvon. Lake Tahoe, California, Dec. 2014, pp. 169–175. URL: https://aclanthology.org/2014.iwslt-papers.3 (visited on 09/19/2024).

[3] E. O. Brigham and R. E. Morrow. "The fast Fourier transform". In: *IEEE Spectrum* 4.12 (Dec. 1967). Conference Name: IEEE Spectrum, pp. 63–70. ISSN: 1939-9340. DOI: 10.1109/MSPEC.1967.5217220. URL: https://ieeexplore.ieee.org/document/5217220 (visited on 09/17/2024).

[4] Seamless Communication et al. *Seamless: Multilingual Expressive and Streaming Speech Translation*. arXiv:2312.05187 [cs, eess]. Dec. 2023. DOI: 10.48550/arXiv.2312.05187. URL: http://arxiv.org/abs/2312.05187 (visited on 03/08/2024).

[5] Tu Anh Dinh and Jan Niehues. *Perturbation-based QE: An Explainable, Unsupervised Word-level Quality Estimation Method for Blackbox Machine Translation*. arXiv:2305.07457 [cs]. July 2023. DOI: 10.48550/arXiv.2305.07457. URL: http://arxiv.org/abs/2305.07457 (visited on 02/16/2024).

[6] Marina Fomicheva et al. "Unsupervised Quality Estimation for Neural Machine Translation". In: *Transactions of the Association for Computational Linguistics* 8 (Sept. 2020), pp. 539–555. ISSN: 2307-387X. DOI: 10.1162/tacl_a_00330. URL: https://doi.org/10.1162/tacl_a_00330 (visited on 03/28/2024).

[7] Markus Freitag et al. "Results of WMT23 Metrics Shared Task: Metrics Might Be Guilty but References Are Not Innocent". en. In: *Proceedings of the Eighth Conference on Machine Translation*. Singapore: Association for Computational Linguistics, 2023, pp. 578–628. DOI: 10.18653/v1/2023.wmt-1.51. URL: https://aclanthology.org/2023.wmt-1.51 (visited on 03/28/2024).

[8] Muhammed Kocyigit, Jiho Lee, and Derry Wijaya. "Better Quality Estimation for Low Resource Corpus Mining". In: *Findings of the Association for Computational Linguistics: ACL 2022*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 533–543. DOI: 10.18653/v1/2022.findings-acl.45. URL: https://aclanthology.org/2022.findings-acl.45 (visited on 08/26/2024).

[9] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. *HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis*. arXiv:2010.05646 [cs, eess]. Oct. 2020. DOI: 10.48550/arXiv.2010.05646. URL: http://arxiv.org/abs/2010.05646 (visited on 05/23/2024).

[10] H.J. Landau. "Sampling, data transmission, and the Nyquist rate". In: *Proceedings of the IEEE* 55.10 (Oct. 1967). Conference Name: Proceedings of the IEEE, pp. 1701–1706. ISSN: 1558-2256. DOI: 10.1109/PROC.1967.5962. URL: https://ieeexplore.ieee.org/document/1447892 (visited on 08/31/2024).

[11] Ngoc-Tien Le, B. Lecouteux, and L. Besacier. "Joint ASR and MT Features for Quality Estimation in Spoken Language Translation". In: Dec. 2016. URL: https://www.semanticscholar.org/paper/Joint-ASR-and-MT-Features-for-Quality-Estimation-in-Le-Lecouteux/d40b93f5352bff348f3bc16cc94b84796bd48df9 (visited on 09/19/2024).

[12] Tomer Levinboim et al. *Quality Estimation for Image Captions Based on Large-scale Human Evaluations*. arXiv:1909.03396 [cs]. June 2021. DOI: 10.48550/arXiv.1909.03396. URL: http://arxiv.org/abs/1909.03396 (visited on 09/06/2024).

[13] Nicolas Morales, Liang Gu, and Yuqing Gao. "Adding noise to improve noise robustness in speech recognition". In: vol. 2. Aug. 2007, pp. 930–933. DOI: 10.21437/Interspeech.2007-335.

[14] Loris Nanni, Gianluca Maguolo, and Michelangelo Paci. "Data augmentation approaches for improving animal audio classification". In: *Ecological Informatics* 57 (May 2020), p. 101084. ISSN: 1574-9541. DOI: 10.1016/j.ecoinf.2020.101084. URL: https://www.sciencedirect.com/science/article/pii/S1574954120300340 (visited on 08/27/2024).

[15] Jan Niehues and Ngoc-Quan Pham. "Modeling Confidence in Sequence-to-Sequence Models". en. In: *Proceedings of the 12th International Conference on Natural Language Generation*. Tokyo, Japan: Association for Computational Linguistics, 2019, pp. 575–583. DOI: 10.18653/v1/W19-8671. URL: https://www.aclweb.org/anthology/W19-8671 (visited on 08/26/2024).

[16] Kishore Papineni et al. "Bleu: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Ed. by Pierre Isabelle, Eugene Charniak, and Dekang Lin. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 311–318. DOI: 10.3115/1073083.1073135. URL: https://aclanthology.org/P02-1040 (visited on 08/30/2024).

[17] Daniel S. Park et al. "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition". In: *Interspeech 2019*. arXiv:1904.08779 [cs, eess, stat]. Sept. 2019, pp. 2613–2617. DOI: 10.21437/Interspeech.2019-2680. URL: http://arxiv.org/abs/1904.08779 (visited on 03/18/2024).

[18] Maja Popović. “chrF: character n-gram F-score for automatic MT evaluation”. In: *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Ed. by Ondřej Bojar et al. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 392–395. DOI: 10.18653/v1/W15-3049. URL: https://aclanthology.org/W15-3049 (visited on 08/30/2024).

[19] Daniel Povey et al., eds. *The Kaldi Speech Recognition Toolkit*. Meeting Name: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society, 2011.

[20] Ricardo Rei et al. *COMET: A Neural Framework for MT Evaluation*. arXiv:2009.09025 [cs]. Oct. 2020. DOI: 10.48550/arXiv.2009.09025. URL: http://arxiv.org/abs/2009.09025 (visited on 07/11/2024).

[21] Nivedita Sethiya and Chandresh Kumar Maurya. *End-to-End Speech-to-Text Translation: A Survey*. en. arXiv:2312.01053 [cs, eess]. June 2024. URL: http://arxiv.org/abs/2312.01053 (visited on 08/26/2024).

[22] Kashif Shah et al. “SHEF-LIUM-NN: Sentence level Quality Estimation with Neural Network Features”. en. In: *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*. Berlin, Germany: Association for Computational Linguistics, 2016, pp. 838–842. DOI: 10.18653/v1/W16-2392. URL: http://aclweb.org/anthology/W16-2392 (visited on 09/06/2024).

[23] Connor Shorten and Taghi M. Khoshgoftaar. “A survey on Image Data Augmentation for Deep Learning”. en. In: *Journal of Big Data* 6.1 (July 2019), p. 60. ISSN: 2196-1115. DOI: 10.1186/s40537-019-0197-0. URL: https://doi.org/10.1186/s40537-019-0197-0 (visited on 08/27/2024).

[24] Julius Smith. *Digital Audio Resampling Home Page*. URL: https://ccrma.stanford.edu/~jos/resample/ (visited on 06/25/2024).

[25] Matthew Snover et al. “A Study of Translation Edit Rate with Targeted Human Annotation”. In: *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*. Cambridge, Massachusetts, USA: Association for Machine Translation in the Americas, Aug. 2006, pp. 223–231. URL: https://aclanthology.org/2006.amta-papers.25 (visited on 04/16/2024).

[26] Lucia Specia et al. “Findings of the WMT 2021 Shared Task on Quality Estimation”. In: *Proceedings of the Sixth Conference on Machine Translation*. Ed. by Loic Barrault et al. Online: Association for Computational Linguistics, Nov. 2021, pp. 684–725. URL: https://aclanthology.org/2021.wmt-1.71 (visited on 09/19/2024).

[27] Matthias Sperber et al. *Evaluating the IWSLT2023 Speech Translation Tasks: Human Annotations, Automatic Metrics, and Segmentation*. arXiv:2406.03881 [cs]. June 2024. DOI: 10.48550/arXiv.2406.03881. URL: http://arxiv.org/abs/2406.03881 (visited on 09/19/2024).

[28] Gabriel Studer et al. “QMEANDisCo—distance constraints applied on model quality estimation”. In: *Bioinformatics* 36.6 (Mar. 2020), pp. 1765–1771. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz828. URL: https://doi.org/10.1093/bioinformatics/btz828 (visited on 09/06/2024).

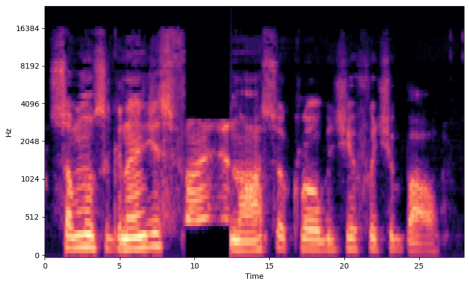
[29] Ioannis Tsiamas et al. *Pushing the Limits of Zero-shot End-to-End Speech Translation.* en. arXiv:2402.10422 [cs]. June 2024. URL: http://arxiv.org/abs/2402.10422 (visited on 08/26/2024).

[30] Yi-Lin Tuan et al. *Quality Estimation without Human-labeled Data.* arXiv:2102.04020 [cs]. Feb. 2021. DOI: 10.48550/arXiv.2102.04020. URL: http://arxiv.org/abs/2102.04020 (visited on 08/26/2024).

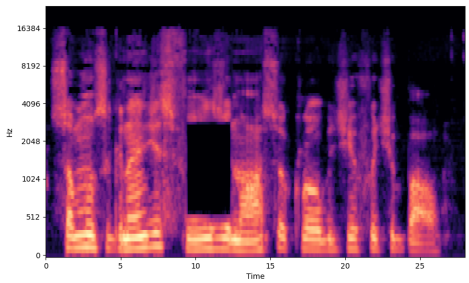
A. Appendix

A.1. Segment-level Perturbation

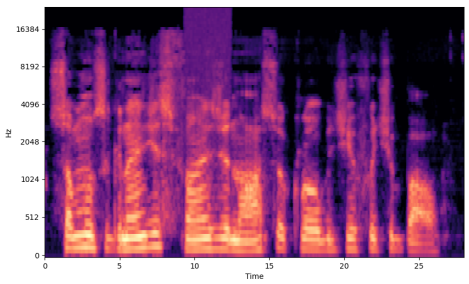
We implement word-level perturbation based on a given transcription of the source language speech by dividing the audio into n pieces where $n-1$ is the number of spaces in the transcription. This trivial audio segmentation aims to approximate a word-boundary identification system. Figure A.1 shows each perturbation strategy applied on the fourth section of speech audio, as per our implementation.



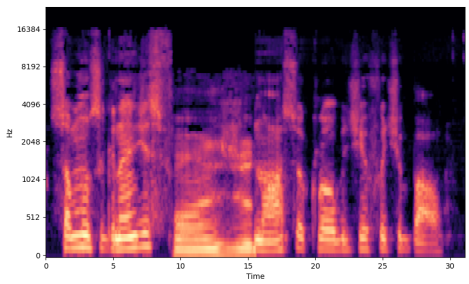
(a) Band pass filter with bounds $[500Hz, 3000Hz]$ on segment 4.



(b) Band stop filter with bounds $[500Hz, 3000Hz]$ on segment 4.



(c) Added Gaussian Noise with a standard deviation of 0.007 to segment 4.



(d) Warping at factor 0.5 on segment 4.

Figure A.1.: Audios with perturbed segments. Each perturbation strategy is applied on the segment.

A.2. Configurations of Exhibited QE-model variants

During our evaluation, we perform multiple ablations and compare various models on performance. Their respective configurations can be found here.

Parameter	PB-QuESTT-chrf*	PB-QuESTT-ter*	PB-QuESTT-bleu*
Metric	CHRF	TER	BLEU
Corpus/Pairwise	Corpuslike	Corpuslike	Corpuslike
Resampling	21kHz, 22kHz, 23kHz, 24kHz, 25kHz	-	-
Warping	-	0.66, 2.0	-
Noising	-	0.1, 0.5, 0.7	0.001, 0.003, 0.005, 0.007
Filtering	-	pass-$[100Hz, 3000Hz]$	-

Table A.1.: Configurations of QE-Model variants using the corpuslike translation similarity calculation strategy.

Parameter	PB-QuESTT-chrf	PB-QuESTT-ter	PB-QuESTT-bleu
Metric	CHRF	TER	BLEU
Corpus/Pairwise	Pairwise	Pairwise	Pairwise
Resampling	-	24kHz, 23kHz, 22kHz	-
Warping	0.66, 2.0	0.7, 2.0, 1.9	-
Noising	0.1, 0.5, 0.7	0.2, 0.3, 0.35	0.001, 0.003, 0.005,0.007
Filtering	pass-$[100Hz, 3000Hz]$	-	-
Weights	evenly	noise-0.3: 0.9, noise-0.2: 0.8, noise-0.35: 0.7, warp-0.7: 0.6, warp-2.0: 0.8, warp-1.9: 0.6, resampling-24000: 0.6, resampling-23000: 0.8	noise-0.001: 0.7, noise-0.003: 0.9, noise-0.005: 1.3, noise-0.007: 1.9

Table A.2.: Configurations of QE-Model variants using the pairwise translation similarity calculation strategy.

Ablation	Metric	C/P	Specs	Weights
Noise only	CHRF	Pairwise	0.1, 0.5, 0.7	evenly
Warp only	CHRF	Pairwise	0.66, 2.0	evenly
Filter only	CHRF	Pairwise	pass-[$100Hz$, $3000Hz$]	evenly

Table A.3.: Configurations of ablations of PB-QUESTT-chrf.

Variant	Metric	C/P	Specs	Weights
Noise best	BLEU	Corpus	0.003, 0.005, 0.007, 0.009, 0.012	-
Resample best	CHRF	Corpus	21kHz, 22kHz, 23kHz, 24kHz, 25kHz	-
Warp best	CHRF	Corpus	0.66, 0.7, 1.9, 2.0, 2.1	-
Filter best	CHRF	Corpus	pass-[$100Hz$, $1100Hz$], pass-[$50Hz$, $1000Hz$], pass-[$500Hz$, $3000Hz$], pass-[$1000Hz$, $7000Hz$], stop-[$2000Hz$, $5000Hz$], stop-[$200Hz$, $1200Hz$], stop-[$100Hz$, $1000Hz$], stop-[$500Hz$, $5000Hz$], stop-[$450Hz$, $5500Hz$], stop-[$550Hz$, $4500Hz$],	-

Table A.4.: Configurations of exhibited QE-Model variants using only a single perturbation strategy.