Code-Mixed Text Augmentation for Latvian ASR

Martins Kronis[†], **Askars Salimbajevs**[†‡], **Mārcis Pinnis**[†‡]
[†]Tilde / Vienības gatve 75A, Riga, Latvia
[‡]Faculty of Computing, University of Latvia / Raiņa bulv. 19, Riga, Latvia
{name.surname}@tilde.lv

Abstract

Code-mixing has become mainstream in the modern, globalised world and affects low-resource languages, such as Latvian, in particular. Solutions to developing an automatic speech recognition system (ASR) for code-mixed speech often rely on specially created audio-text corpora, which are expensive and time-consuming to create. In this work, we attempt to tackle code-mixed Latvian-English speech recognition by improving the language model (LM) of a hybrid ASR system. We make a distinction between inflected transliterations and phonetic transcriptions as two different foreign word types. We propose an inflected transliteration model and a phonetic transcription model for the automatic generation of said word types. We then leverage a large human-translated English-Latvian parallel text corpus to generate synthetic code-mixed Latvian sentences by substituting in generated foreign words. Using the newly created augmented corpora, we train a new LM and combine it with our existing Latvian acoustic model (AM). For evaluation, we create a specialised foreign word test set on which our methods yield up to 15% relative CER improvement. We then further validate these results in a human evaluation campaign.

Keywords: code-mixing, speech recognition, inflected transliteration, parallel data, language model

1. Introduction

It has become increasingly popular for Latvian (L1) speakers to use English (L2) words in their speech. This is especially the case for younger generations or among technical field experts, where the former consume more and more of completely English-dominated multimedia and the latter struggle with frequent absence (or lack of knowledge) of highly specialised Latvian terms within the specific field.

During speech, word substitution is performed by either pronouncing the actual English word or by adapting it to the Latvian grammar rules (typically by adding endings and inflexions). We will use the term "phonetic transcription" for the first case, and call the latter "inflected transliteration" because new loanwords are being coined. In both cases, recognition of said words poses a challenge to automatic speech recognition (ASR) systems, especially hybrid ones, which consist of an acoustic model (AM) and a language model (LM) and remain popular in the industry. Even if the AM manages to correctly recognise foreign phonemes (which is possible if there is a significant overlap between L1 and L2 phonemes), the LM will still struggle with both inflected transliterations and phonetic transcriptions, as they are poorly represented in most training corpora. This results in low context-level probabilities and leads to the rejection of the correct hypotheses during decoding. The same data sparsity problem also applies to end-to-end neural network ASR models, though character or subword end-to-end models are typically much better at producing unseen words.

Probably the best possible solution would be the creation of new training corpora which cover both phenomena. Ideally, this would also include audio data for AM training. However, this is a very time-consuming and expensive process. Moreover, there will always be some highly specialised field terms which the existing data will not cover.

We, therefore, focus our attention on improving the language model performance by proposing a data augmentation method for enriching Latvian LM training corpora with automatically generated English phonetic transcriptions and/or inflected transliterations. More specifically, two models are developed - a model for inflected transliteration generation, as well as a model for phonetic transcription generation. We then propose a pipeline for said model usage on a parallel English-Latvian corpus to perform contextually aware substitutions of inflected transliterations and/or phonetically transcribed words. The resulting augmented corpus is used to train a new LM, which demonstrates consistent improvement in foreign word recognition in a hybrid ASR system.

The problem is not unique to Latvian. The same problem applies to Lithuanian, Estonian, and many other languages. Therefore, while in this paper we focus on English and Latvian only, we believe that our results and proposed method can be useful for other language pairs.

2. Related Work

Code-switching in ASR has been an active research area for many years. The first efforts included a multi-pass approach (Lyu and Lyu, 2008),

LREC-COLING 2024, pages 3469–3479

where language boundary detection (LBD) is used to divide the input utterance into segments that are language-homogeneous. Then, a corresponding language-dependent ASR model can be used on each segment. This approach has a number of limitations: the performance is bounded by language identification accuracy, segmentation introduces context-breaking and degrades recognition quality, and, finally, it requires training and running several independent ASR models.

Researchers then opted for a one-pass approach, where one common ASR system - is capable of recognising such mixed speech. This approach necessitates the creation of relevant language resources and code-switched speech data. For example, Hamed et al. (2022) addressed the code-switching ASR by creating a code-switched speech corpus for training models for dialectal low-resource Arabic-English speech recognition.

However, the manually annotated code-mixed data is frequently insufficient to train a high-quality ASR model. Therefore, researchers investigate automatic data augmentation with foreign words, e.g. Yu et al. (2023) explore using parallel Mandarin-English text to perform English word substitutions based on Mandarin word frequency. Synthetic parallel data is obtained using neural-machine-translation (NMT) and word alignments are estimated using statistical methods. Text-only data is then injected into an end-to-end (E2E) model by cross-modality learning or is used in combination with Text-to-Speech (TTS) to create a synthetic audio-text corpus. Experiments show substantial **r**WER improvements over non-augmented baseline E2E models.

Similarly, Punjabi et al. (2019) explore code-mixed text corpus generation from a synthetic parallel English-Hindi corpus obtained using NMT, but propose word alignment estimation based on the NMT model's attention weights. Substitution is then performed based on Hindi word frequency. Results show perplexity and word error rate improvements for an augmented hybrid-ASR system using a DNN-HMM acoustic model and an n-gram LM.

Finally, Pratapa et al. (2018) also use NMT to generate a parallel text corpus, but synthesise the code-mixed corpus on sentence level, by leveraging Equivalence Constraint theory. Their experiments show a reduction of perplexity for the newly trained recurrent neural network (RNN) LMs.

While previous studies on code-mixed text augmentation have made use of NMT-generated parallel corpora (Yu et al., 2023; Punjabi et al., 2019; Pratapa et al., 2018), we improve on that by utilising a large human-translated parallel English-Latvian corpus. Furthermore, we observed that proposed word substitutions typically involve only the original word or occasionally its phonetic transcription. In our study, we perform substitutions using inflected transliterations and phonetically transcribed words. Finally, it should be noted that most of the previous work performed word substitution based on word frequency in the target (L1) language (e.g., Hindi in Punjabi et al., 2019). Since we are using parallel corpora individual word frequencies across L1 and L2 are almost identical and as we have a more robust English word IDF resource available, we instead focus on identifying substitution pairs based on word frequencies in the source, English, language.

3. Method

We begin this section by describing the hybrid ASR system that will be used to carry out the experiments. We then focus our attention on the two proposed models – the inflected transliteration model and the phonetic transcription model. Finally, we introduce the pipeline for LM training data augmentation by leveraging parallel data and the two proposed models.

3.1. Hybrid ASR System

Our baseline Latvian ASR system is a Kaldi (Povey et al., 2011) based hybrid ASR system, which uses a sub-word approach to tackle the vocabulary size problem of a highly inflective language. It is designed to recognise sequences of right-marked BPE sub-word units following the approach by Smit et al. (2017), which then are reconstructed into full words during post-processing.

The acoustic model is a hybrid hidden Markov and a Time-delay deep neural network (TDNN) model, which is trained on LSRC (Pinnis et al., 2014), LSDC (Pinnis et al., 2016) and SAEIMA (Salimbajevs, 2018) speech corpora (about 300h in total, $\approx 1,500$h after speed and reverb augmentation (Ko et al., 2015)). The model is trained to output Latvian letters directly instead of phonemes, as it was empirically found that such an approach improves both accuracy and robustness.

Our baseline uses a 4-gram model trained for specially pre-processed texts for language modelling. It is filtered to contain only characters from the Latvian alphabet, punctuation removed and all words are lowercased. Numbers are converted to words, while abbreviations are spelled out using a previously constructed abbreviation dictionary. Finally, word splitting is performed using byte-pair encoding (BPE). This allows the model to cover all possible word inflexions and variations with a fixed vocabulary.

Since our proposed method is text augmentation for LM, the hybrid ASR approach seems a natural choice, as it allows to isolate the language modelling component. Also since Latvian is a limited resource language, training a heavier Transformer-based hybrid architecture or a full E2E system, has proven to be challenging and not always results in outperforming Kaldi-based system.

We also opted out of training RNN or Transformer models as initial testing suggested only minor differences with respect to n-gram models, and the result was not worth the increased training and test times.

3.2. Inflected Transliteration Model

Although some transliteration models are presented in the scientific literature (e.g. Le and Sadat, 2017; Kundu et al., 2018), they generally fail to consider word inflexions. Furthermore, there are no models developed specifically for the English-to-Latvian case. In this subsection, we propose an English-to-Latvian transliteration model capable of generating inflected transliterations according to Latvian grammar.

Nouns, pronouns and adjectives in Latvian have seven cases: nominative, genitive, dative, accusative, instrumental, locative, and vocative. Additionally, their inflexion depends on the number (singular or plural) and gender, of which there are two. The Latvian verbal morphology is even more complex, as verbs are organised in five conjugation classes having six tenses, five moods and two voices.

A desired transliteration process would, therefore, include inflecting the English source word the same way the target Latvian word (which is being replaced) was inflected in a given sentence. An example is provided in Table 1.

Most of the information necessary for correct inflexion can be gathered from the original target word itself, but some ambiguities can only be resolved by providing context. One option would be to train a sentence-level model, but that would require relatively large amounts of data and result in a bigger and slower model. Additionally, it would constrain the model input to sentences, which is undesirable if the end goal is a single word-to-word transliteration. Instead, we train a character-level model on target-source (L2-L1) word pairs and provide (along with the source word in input) a part-of-speech (POS) tag of the expected inflexion of the target (Latvian) word by utilising our existing POS tagger. For Latvian, tags are positional, consisting of 28 characters describing the full morpho-syntactic description (e.g. number, person, gender, etc.) of the word. This necessitates the creation of a Latvian-English inflected transliteration dictionary to be used as training data.

3.2.1. Inflected transliteration Dictionary

In order to extract the inflected transliterated dictionary, we used a proprietary parallel English-Latvian corpus, which is typically used to train neural machine translation systems. The inflected transliteration word dictionary was created as follows:

1. First, the corpus was cleaned, normalised, filtered, and pre-processed (tokenised and truecased) using methods and tools from Pinnis et al. (2018).

2. Then, both English and Latvian sides were lemmatised using part-of-speech taggers for Latvian and English. We used averaged perceptron classifiers by Nikiforovs[1].

3. After lemmatization, we performed word alignment using eflomal (Östling and Tiedemann, 2016).

4. Then, we extracted source-to-target and target-to-source lexical translation tables using the word alignments with the help of scripts from the Moses toolkit (Koehn et al., 2007).

5. Finally, we filtered resulting noisy probabilistic dictionaries, using the *FilterGizaDictionary*[2] (Aker et al., 2014) tool. This tool filters noise (e.g., stop-words paired with content words, punctuation or numerals paired with words, etc.) and as a side-product produces inflected transliteration dictionaries. A pair of source and target words are assumed to be transliterations if both words are longer than three letters and they have a similarity score greater or equal to 0.7. To calculate similarity, we transform the Levenshtein distance (Levenshtein, 1966) between stemmed variants of the two words into a similarity metric.

3.2.2. Architecture and Training

We trained machine inflected transliteration systems using the Marian neural machine translation (NMT) toolkit (Junczys-Dowmunt et al., 2018). The models are Transformer Base (Vaswani et al., 2017) neural networks with six encoder and six decoder layers and tied embeddings. We apply a dropout of 0.1 between the transformer layers, the attention layers, and the feed-forward layers. For training, we use the Adam (Kingma and Ba, 2015) optimiser ($\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 1e-9$) and apply gradient norm clipping to 5. We use a learning rate of $1e-4$ and apply a learning rate warmup of 16000 updates, and then decrease it using inverse square root scheduling. For decoding, we apply a

[1] https://github.com/pdonald/latvian

[2] https://github.com/pmarcis/dict-filtering

Dict. form (LV)	Example inflexion	Inflected form (LV)	Dict. form (EN)	Translit. (LV)
nožēlot	verb, present, third person, indicative	nožēlo	regret	regretē
iedomāties	participle, past, masculine, sing., nominative, indefinite adjective, active voice, reflexive	iedomājies	imagine	imaginējies
skaidrs	adjective, masculine, plural, nominative, comparative degree, indefinite adjective	skaidrāki	improved	impruvētāki

Table 1: Example of transliterations with inflections from English to Latvian

maximum target length factor of 1.5 and normalise the translation score with a coefficient of 0.6.

In our experiments, the maximum input sequence length was set to 25. This allowed us to cover 99.9% of input data. The vocabulary of the machine inflected transliteration system consists of 38 letters, the apostrophe symbol (for English words), 865 different morpho-syntactic tags (for various inflexions of words in Latvian), the unknown token symbol '<*unk*>' and the sentence (i.e., word for machine inflected transliteration) ending symbol '</*s*>'.

Considering that the task is simpler than translation (e.g., the entire vocabulary contains only 906 entries, the sequence lengths are rather small, and inflected transliteration may require only subtle changes to the input), we performed validation after 100 updates (instead of 1000 to 10000 typically applied for translation), and applied early stopping after failing to improve the cross-entropy loss for ten consecutive validation steps. Furthermore, we used a workspace of 4500 MB and allowed Marian to determine the mini-batch size automatically so that it fits in the workspace. All other parameters were default parameters of the Marian NMT toolkit[3].

3.3. Phonetic Transcription Model

For phonetic transcription generation, we propose a model for mapping English words to Latvian via phoneme conversion. An alternative (worse) solution would be to augment the training corpus by substituting L1 (Latvian) words with the actual L2 (English) words. However, this poses a challenge for the AM, which is able to recognise only words written in the Latvian alphabet. This means that it won't be able to output words with English letters "q", "x", "y", and "w". Fine-tuning the acoustic model would require English or code-mixed audio-text corpus and would defeat the general aim of this study. Therefore, we opt for using phonetic transcription instead, allowing the AM to operate on L1 (Latvian) phonemes. This means that the produced sub-words will be also more likely to have overlap with existing L1 vocabulary. If sub-word level overlap is high, this also has the added benefit of improving the generalisation capability of the model.

3.3.1. English-IPA-Latvian Phoneme Mapping

To the best of our knowledge, there are no available complete English-to-Latvian phoneme maps. Furthermore, the overlap between said phonemes is low, as evident from IPA tables. One possible solution is to learn and perform the mapping implicitly by training a neural model on word-transcription pairs. This, however, would necessitate the creation of a phonetically transcribed word dictionary, similar to the one described in 3.2.1, but the phonetic transcriptions would have to be generated manually. A more appealing approach is to try to make use of an intermediary phoneme set, such as the International Phonetic Alphabet (IPA) (Smith, 2000) and pursue a pivot-based conversion - English-to-IPA-to-Latvian.

Mappings for English-to-IPA are well documented and exist as numerous internet resources, with the most complete one available as a help resource for transcription creation in Wikipedia[4].

Somewhat contrary, there are no complete mappings or tools available for Latvian-to-IPA, but a

[3] For precise default parameters, refer to Marian code base at the commit 4dd30b5065efba61fc044e9dc4303205c9d2ac53

[4] https://en.wikipedia.org/wiki/Help:IPA/English

comprehensive compilation based on the work by Nau (1998) is also available as a Wikipedia help resource[5].

When comparing both mappings on the IPA level, it becomes clear that the overlap is not complete. We, therefore, propose our own additions to form a complete English-to-IPA-to-Latvian mapping, which we published on GitHub[6].

3.3.2. Model Implementation

There are numerous rule-based and neural network solutions (e.g. Rosca and Breuel, 2016) available for English-to-IPA conversion. In this work, we will adopt a rule-based solution due to its ease of use and faster overall development time. In particular, for English-to-IPA conversion we will make use of *eng_to_ipa*[7] python package. Conversion from the generated IPA phonemes to Latvian can then be performed using a simple look-up table, based on our English-to-IPA-to-Latvian chart. A conceptual example of the proposed model is shown in Table 2.

Input (EN)	IPA	Output (LV)
moonlight	ˈmunˌlaɪt	mūnlait
phonetics	fəˈnɛtɪks	fanetiks
explanation	ˌɛkspləˈneɪʃən	eksplaneišen

Table 2: Example of phonetic transcription model inputs and outputs

3.4. Corpus Augmentation Pipeline

Our starting point is aligned hand-translated parallel English-Latvian (L2-L1) corpus, with the goal of creating a Latvian corpus for LM training augmented with inflected transliterations and/or phonetic transcriptions. By performing word substitutions on aligned parallel word pairs, we solve the translational ambiguity that can arise from having just the target word and using only a dictionary. This results in lower context-breaking for our synthetic code-mixed sentences. With this in mind, the proposed augmentation pipeline can be summarised as follows:

1. **Extract words of interest (candidate source words):** In order to compile a list of candidate L2 words for inflected transliteration or phonetic transcription, we start by acquiring an inverse document frequency (IDF) table using the L2 side of the parallel corpus. We treat each sentence as a separate document when calculating the IDF scores. We filter out all exceedingly rare words (e.g., named entities, misspelt words, foreign words, etc.), as those are unlikely to appear in the spoken language of L1. This is done by discarding all words with IDF ≥ 12.5. Similarly, we are interested in removing overly common (functional) words (such as prepositions, conjunctions etc.) as those have either been already assimilated into Latvian or are not usually substituted. This is done by applying a stop-word filter. For this, we use the stop-word file published by Pinnis et al. (2012). A similar result can also be achieved by applying an upper bound on the word IDF scores.

2. **Filter parallel corpus based on alignments:** Both inflected transliteration and phonetic transcription models are designed to work on one-to-one word pairs, so using forward and backward alignments, we discard all sentences that have one-to-many word alignments. Performing substitution on such words would also increase the possibility of context breaking and is, therefore, overall undesirable.

3. **Generate POS tags for L1**: POS tags are required as input for the inflected transliteration model and are therefore generated for Latvian (L1) sentences using our proprietary POS tagger. This step can be omitted when choosing to use only the phonetic transcription model for corpus augmentation.

4. **Extract source-target word pairs from parallel corpus**: We match candidate L2 words from step 1 to their target counterparts in the filtered parallel corpus using alignments. The matched pairs are stored alongside the L1 POS tag and the corresponding L1 sentence for later substitution.

5. **Filter source-target word pairs using Levenshtein distance:** In order to avoid re-transliterating or re-transcribing words that already have an existing transliterated/transcribed version in Latvian, we filter the extracted L2-L1 word pairs based on a similarity score. This is done by calculating the Levenshtein distance on stemmed pairs, and all pairs with the similarity score ≥ 0.7 are discarded.

6. **Generate inflected transliterations/phonetic transcriptions:** Cleaned L2-L1 pairs are fed into inflected transliteration and/or phonetic transcription models to generate words for corpus augmentation.

7. **Filter transliterated output:** Inflected transliteration model output is re-cleaned by once

[5] https://en.wikipedia.org/wiki/Help:IPA/Latvian

[6] https://anonymous.4open.science/r/code-mixed-ASR-26DC/tools/ipa2lv_alphabet.yaml

[7] https://github.com/mphilli/English-to-IPA

again calculating stemmed Levensthein distance. This time, we discard all pairs that have a similarity score of ≤ 0.5, with the aim of removing bad transliterations. This step should be omitted for the phonetic transcription model outputs as there is generally lower character overlap between the source and phonetically transcribed words. Additionally, the phonetic transcription model is not a neural network, and the quality of its outputs does not vary but simply reflects the quality of the phoneme mappings.

8. **Substitute generated words back into L1 corpus:** Foreign word component of the filtered L1-transcription/transliteration pairs is substituted back into the corresponding L1 sentences that were obtained from the parallel corpus. As there is usually more than one possible substitution per sentence, one can either sample substitutions according to some probability distribution or perform one substitution per sentence. In our experiments, we focus on the latter, as sampling and performing multiple substitutions per sentence did not show substantial improvements during initial testing. This means that, for example, for an L1 sentence that has three possible distinct substitutions, we create three copies of it in the augmented corpus, each containing one of the respective substitutions.

3.5. Evaluation Corpus

While both inflected transliterations and phonetic transcriptions are very common in spontaneous and colloquial speech, such datasets are quite rare, which is a significant challenge not only for the training of any models but also for testing. Moreover, such a dataset should contain both audio and text.

For example, most of the popular, publicly available Latvian corpora, such as Mozilla Commonvoice by Ardila et al. (2020), have low representation of both inflected transliterations and phonetic transcriptions and do not provide the respective annotations. We, therefore, set out to create our own small testing corpus, with the aim of having substantial representation of both inflected transliterations and phonetic transcriptions, at least one (but not necessarily both) per utterance.

We use our internal researcher meeting recordings as a source of code-mixed and transliterated speech. These recordings often reference domain terminology, papers, conferences or publications in English. We started with three hours of internal meeting audio that is deemed fit for public release and extracted utterances of at least 3 seconds in length. This resulted in 210 final utterances totalling a little over 13 minutes of speech, with each utterance containing at least one phonetically transcribed or transliterated word. The created corpus contains ≈ 9000 characters and roughly 1800 words. There are 178 phonetic transcriptions and 123 inflected transliterations, or 9.8% and 6.8% of total words, respectively.

As annotating phonetic transcriptions is an inherently subjective task, we create two versions of the new corpus - one where English words are left in English, named *text_f_eng*, and one where they are subjectively transcribed to the best of the annotator's judgement, named *text_f*. Inflected transliterations are left untouched as they are, after all, English words that have been adapted into Latvian. The character count between the two corpora remains almost the same. Both versions of the code-mixed corpus are made available in the previously mentioned Github repository.

4. Experimental Setup

This section describes the performed experiments and their respective setups. We performed one experiment per each of the proposed data augmentation methods and one for the combination. The overall aim is to test improvements in foreign word (i.e. inflected transliteration and phonetic transcription) recognition in ASR by retraining the LM using an augmented corpus.

4.1. LM Corpus Augmentation

We use a proprietary 19 M sentence English-Latvian parallel corpus as a starting point. We apply the data augmentation pipeline described in Subsection 3.4 using both inflected transliteration and phonetic transcription models.

To keep the experiments consistent across methods, we first performed inflected transliteration generation, which involves several extra filtering steps, and then ran the phonetic transcription model on the filtered source words from step 7 in the pipeline. This guarantees that the augmented corpora produced by each of the methods have the same unique source words and corresponding parallel sentences. Then, the only difference between them is the nature of the substituted words, i.e. they are generated by either the phonetic transcription model or the inflected transliteration model.

After step 7 of the augmentation pipeline, we were left with ≈ 1.6 M unique substitution word pairs and almost 6 M unique base sentences (i.e. sentences with available substitution pairs). Performing the substitution brought this number to a total of 18.5 M training sentences for each of the data-augmented corpora. This means that roughly 2/3 of training sentences in each of the corpora differ only by a single word. We add an additional

Original sentence	Substitution pair	Augmented sentence
Vērtībai nav **piešķirts** noklusējums	piešķirts - asignēts *(inflected transliteration)*	Vērtībai nav **asignēts** noklusējums
Vērtībai nav **piešķirts** noklusējums	piešķirts - asaind *(phonetic transcription)*	Vērtībai nav **asaind** noklusējums

Table 3: Examples of training sentence augmentation with English word "assigned" using both of the proposed methods. The original sentence translates to "Value has no default assigned."

	# substit. words	# base sent.	# sent. with substitutions	# total sent.
Baseline	-	6 M	-	14 M
Transliterated	1.6 M	6 M	18.5 M	26.5 M
Transcribed	1.6 M	6 M	18.5 M	26.5 M
Combined	3.2 M	6 M	37 M	45 M

Table 4: Summary of created augmented corpora for LM training

8 M sentences non-augmented from the Latvian WebNews corpus, bringing the total training sentence count to 26.5 M. This is done to bolster the training corpus with non code-mixed sentences and increase domain coverage. An example of the enriched training sentences is shown in Table 3 and a detailed summary of the produced corpora is available in Table 4.

4.2. Baseline

To test the success of our approach in improving foreign word recognition, we created a baseline non-augmented training corpus. To ensure that the baseline corpus has the same language coverage, we use the same previously mentioned 6 M base sentences and the same 8 M sentences from the Latvian WebNews corpus. This brings the total unique sentence count of the baseline LM training corpus to 14 M. While this is significantly lower than the augmented corpora, the general language coverage remains about the same, as the majority of the new sentences in the enriched corpora only differ by a single word.

4.3. LM Training

We train a total of four LMs - one for each of the corpora described in Table 4. Training is consistent with the setup described in Subsection 3.1 and features a 4-gram model. We opted out of training RNN LM models as initial testing suggested only minor differences with respect to n-gram models, and the result was not worth the increased training and test times.

4.4. Decoding & Evaluation

To complete the experimental setup, we combine the trained LMs with the AM described in Subsection 3.1, thus forming four complete hybrid ASR systems for Latvian - *baseline*, *transliterated*, *transcribed* and *combined*.

We then performed the following experiments:

- **Automatic evaluation of the overall ASR quality**: The newly created foreign word evaluation corpora described in Subsection 3.5 were decoded using each of the ASR systems. Here, we focused on mainly %CER metric, as the word error rate for a corpus of $\approx$ 1800 total words can be considered too inaccurate.
- **Semi-automatic evaluation of only inflected transliteration and phonetically transcribed word ASR quality**: We manually extracted the foreign words from both *test_f* and *test_f_eng* corpora and aligned them with the produced hypotheses.%CER was then calculated on the individual aligned words. Results were considered by foreign word type, i.e. looking at inflected transliterations and phonetic transcriptions separately. This experiment was designed to examine performance improvements in greater detail. Additionally, we aimed to highlight the differences in phonetic transcription and inflected transliteration recognition between data augmentation methods.
- **Semi-automatic evaluation of other word ASR quality**: We removed the manually extracted foreign words from the produced hypotheses and respective references. %CER was then calculated to look at performance degradation in general language.
- **Manual evaluation of only inflected transliteration and phonetically transcribed word ASR quality**: Recognising the limitations of a small testing corpus, we also performed a manual blinded evaluation on the individually extracted foreign words. The foreign word hypotheses from the three data-augmented mod-

els, namely *transliterated*, *transcribed*, and *combined*, were compared against the baseline hypothesis (but not the reference). Each comparison was then judged to be strictly better than the baseline, strictly worse or the baseline was deemed good enough from the start. For evaluation, we employed two Latvian native speakers with expert background in linguistics.

5. Results & Discussion

We first discuss the results obtained from decoding the full evaluation set, shown in Table 7.

For both versions of the dataset, all of the proposed augmentation methods show minor improvements in character error rate, with the largest increase of 1.4 %**r**CER when using phonetic transcription augmentation on *test_f*. An exception to this is the *combined* augmentation, which is worse than the baseline on *test_f_eng* corpus. Despite being task-specific, both corpora contain only a little over 16% foreign words, and thus, low absolute and relative %CER changes are expected.

We instead focus our attention on Table 5, which shows more detailed breakdown of foreign word recognition performance.

It is apparent that *transcribed* language model shows the most consistent and substantial improvements across all foreign word types. It is 1.6 %**r**CER better than baseline on inflected transliterations and 15.4 %**r**CER better on phonetic transcriptions. The improvement remains substantial also for English words, roughly 10.0 %**r**CER, which suggests that the said augmentation method is fairly stable to annotator subjectivity in creating phonetic transcriptions. When considering the general domain, the performance degradation compared to the baseline is small, $\approx$ 0.7 absolute %CER and 3.6 %**r**CER.

Table 5 also shows that *transliterated* language model yields consistent, but minor (1.4 %**r**CER - 2.8 %**r**CER) improvements over the baseline on all foreign word types. As expected, the performance on phonetic transcriptions is significantly lower when compared to *transcribed* method. However, it is also interesting to observe that despite including inflected transliterations in the training corpus, this model does not outperform *transcribed* method on inflected transliteration recognition. Additionally, there is no degradation in the general domain when compared to the baseline. This result was rather unexpected and warrants further more extensive investigation. A possible explanation could be low frequency of inflected transliteration subwords, which could then be alleviated by retraining the BPE model.

Finally, when considering the *combined* method, Table 5 shows that the method mimics *transcribed* model performance on phonetic transcriptions but exhibits larger degradation of 5.8 %**r**CER on general domain words. Additionally, it is the only LM to show degradation on inflected transliterations, performing roughly 4.4 %**r**CER worse than the baseline. Being a combination of *transliterated* and *transcribed* augmentations, it likely inherits performance advantages or disadvantages from both models. At 45 M sentences, *combined* augmented corpus could also be considered too large for 4-gram model and could benefit from a higher order model or a change to RNN architecture.

Moving on to subjective evaluation presented in Table 6, we can see that both annotators strongly agree on *transcribed* and *combined* models providing significant improvements in phonetic transcription recognition, recognising up to 18 % foreign words better than the baseline, and only around 3 % worse. Annotators also agree on model augmented with *transliterations* providing modest improvement on phonetic transcriptions.

However, there is no strong agreement on inflected transliterations. While first annotator tended to prefer the baseline a bit more, the second annotator found data augmented models to be better.

Subjective evaluation results support the previous objective findings, we see consistent and significant improvement on phonetic transcriptions, as well as small or non-existent improvement on inflected transliterations. Both evaluations agree that model augmented with transcriptions only is the best model among the tested.

6. Limitations

A general concern with synthetic data, such as generated foreign words, is potentially poor approximation of real-life domain. For example, not all of our generated foreign words are used in real code-mixed speech. Additionally, some foreign words may be generated in a form that matches an existing L1 word which can affect the recognition performance of contexts where said word is used. Furthermore, since foreign word spelling and transliteration inflection can differ between speakers, it is possible that synthetic foreign words do not provide a 1:1 match to the reference or even a real life scenario, resulting in overall lower precision of the system. As discussed previously, we addressed these limitations by sampling candidate words from a large natural speech dataset, using post-generation filtration, as well as using human evaluation to account for possible spelling differences between the reference and the hypothesis.

Another limitation concerns usage of English transcriptions for data augmentation instead of real

	Infl. transliterations	Phonetic transcriptions	English	Other words (LV)
Baseline	42.51	61.58	71.86	**18.85**
Trasliterated	41.92	59.88	70.63	**18.85**
Transcribed	**41.82**	**52.07**	**64.69**	19.54
Combined	44.37	52.21	65.10	19.96

Table 5: %CER for the four methods on manually extracted and aligned foreign words. The tab **other words** shows the performance on *test_f* corpus with manually removed foreign words from both the references and the hypotheses. Entries in **bold** highlight the best results; numerically lower results are better.

		Transliterated		**Transcribed**		**Combined**	
		transl.	phon. trsc.	transl.	phon. trsc.	transl.	phon. trsc.
A-1	worse	16 [13.0%]	5 [2.8%]	13 [10.6%]	4 [2.3%]	19 [15.5%]	5 [2.8%]
	better	14 [11.4%]	9 [5.1%]	13 [10.6%]	29 [16.3%]	14 [11.4%]	30 [16.8%]
A-2	worse	6 [4.9%]	5 [2.8%]	7 [5.7%]	3 [1.7%]	12 [9.8%]	6 [3.4%]
	better	12 [9.8%]	12 [6.7%]	14 [11.4%]	27 [15.2%]	15 [12.2%]	32 [18.0%]

Table 6: Total and relative amount of foreign words judged to be better or worse than the baseline for the three proposed augmentation methods. Tab *transl.* refers to inflected transliterations, and *phon. trsc.* to phonetic transcriptions. Inter annotator agreement between **annotators A-1 and A-2** is measured using Free Marginal Kappa (Randolph, 2005), $\kappa = 0.836$. Samples were baseline was judged to be good enough and no positive or negative changes were present in the augmented model hypotheses are excluded from the table.

	test_f	test_f_eng
Baseline	25.43	27.45
Transliterated	25.27	**27.29**
Transcribed	**25.08**	27.41
Combined	25.34	27.62

Table 7: %CER for the four methods on the full test corpora. Entries in **bold** highlight the best results; numerically lower results are better.

English words, which leads to the ASR system also outputting transcriptions. This is typically not the desired behaviour, especially during deployment. We addressed the reasons behind this choice in Subsection 3.3. However, this issue can be easily resolved by adding a post-processing step that maps the output phonetically transcribed English words back to their true English form using our proposed English-IPA-Latvian phoneme sets.

7. Conclusion

In this work, we presented a solution to boosting foreign (English) word recognition in code-mixed Latvian speech by augmenting the LM training corpus of a hybrid-ASR Latvian system. We identified suitable substitution pairs in parallel EN-LV textual data and replaced the Latvian word with a foreign word generated by our proposed inflected transliteration or phonetic transcription models. The resulting augmented text corpora were used to train new Latvian LMs that were evaluated in a hybrid ASR setup. We calculated %CER on the specially created code-mixed corpus. For the LM trained using the transcription augmentation method, we observed a substantial 15.4% relative CER improvement in phonetic transcription recognition as well as a small, but consistent improvement in inflected transliteration recognition. The model trained using inflected transliteration augmentation yielded smaller improvements and was outperformed by the *transcribed* model in recognition of both foreign word types. The combined augmentation method model was not found to produce any improvements and performed worse than the respective individual models. The consistency of these findings was reinforced by human evaluation and testing on English (not phonetically transcribed) words.

8. Acknowledgements

This research has been supported by the ICT Competence Centre (www.itkc.lv) within the project *2.2 Mākslīgais intelekts reālā laika subtitrēšanai un dublēšanai tiešraidēm* (2.2 Artificial intelligence for real time subtitling and dubbing in live streams) of EU Structural funds, ID no 5.1.1.2.i.0/1/22/A/CFLA/0083.

9. Bibliographical References

Ahmet Aker, Monica Paramita, Mārcis Pinnis, and Robert Gaizauskas. 2014. Bilingual dictionaries for all eu languages. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222.

Injy Hamed, Pavel Denisov, Chia-Yu Li, Mohamed Elmahdy, Slim Abdennadher, and Ngoc Thang Vu. 2022. Investigations on speech recognition systems for low-resource dialectal arabic–english code-switching speech. *Computer Speech & Language*, 72:101278.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, et al. 2018. Marian: Fast neural machine translation in c++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. 2015. Audio augmentation for speech recognition. In *Proc. Interspeech 2015*, pages 3586–3589.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Soumyadeep Kundu, Sayantan Paul, and Santanu Pal. 2018. A deep learning based approach to transliteration. In *Proceedings of the Seventh Named Entities Workshop*, pages 79–83, Melbourne, Australia. Association for Computational Linguistics.

Tan Le and Fatiha Sadat. 2017. A neural network transliteration model in low resource settings. In *Proceedings of Machine Translation Summit XVI: Research Track*, pages 337–345, Nagoya Japan.

Vladimir Iosifovich Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

Dau-Cheng Lyu and Ren-Yuan Lyu. 2008. Language identification on code-switching utterances using multiple cues. In *Proc. Interspeech 2008*, pages 711–714.

N. Nau. 1998. *Latvian*. Languages of the world / Materials: Materials. LINCOM EUROPA.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *Prague Bulletin of Mathematical Linguistics*, 106:125–146.

Mārcis Pinnis, Andrejs Vasiļjevs, Rihards Kalniņš, Roberts Rozis, Raivis Skadiņš, and Valters Šics. 2018. Tilde MT platform for developing client specific MT solutions. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Mārcis Pinnis, Ilze Auziņa, and Kārlis Goba. 2014. Designing the Latvian Speech Recognition Corpus. In *Proceedings of the 9th edition of the Language Resources and Evaluation Conference (LREC'14)*, pages 1547–1553.

Mārcis Pinnis, Radu Ion, Dan Ştefănescu, Fangzhong Su, Inguna Skadiņa, Andrejs Vasiļjevs, and Bogdan Babych. 2012. ACCURAT Toolkit for Multi-Level Alignment and Information Extraction from Comparable Corpora. In *Proceedings of the ACL 2012 System Demonstrations*, pages 91–96. Association for Computational Linguistics.

Mārcis Pinnis, Askars Salimbajevs, and Ilze Auzina. 2016. Designing a Speech Corpus for the Development and Evaluation of Dictation Systems in Latvian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553, Melbourne, Australia. Association for Computational Linguistics.

Surabhi Punjabi, Harish Arsikere, and Sri Garimella. 2019. Language model bootstrapping using neural machine translation for conversational speech recognition. *CoRR*, abs/1912.00958.

Justus J. Randolph. 2005. Free-Marginal Multirater Kappa (multirater K[free]): An Alternative to Fleiss' Fixed-Marginal Multirater Kappa. *Joensuu Learning and Instruction Symposium*.

Mihaela Rosca and Thomas M. Breuel. 2016. Sequence-to-sequence neural network models for transliteration. *CoRR*, abs/1610.09565.

Askars Salimbajevs. 2018. Creating Lithuanian and Latvian Speech Corpora from Inaccurately Annotated Web Data. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA).

Peter Smit, Sami Virpioja, Mikko Kurimo, et al. 2017. Improved subword modeling for wfst-based speech recognition. In *INTERSPEECH*, pages 2551–2555.

Caroline L. Smith. 2000. Handbook of the international phonetic association: a guide to the use of the international phonetic alphabet (1999). *Phonology*, 17(2).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Haibin Yu, Yuxuan Hu, Yao Qian, Ma Jin, Linquan Liu, Shujie Liu, Yu Shi, Yanmin Qian, Edward Lin, and Michael Zeng. 2023. Code-Switching Text Generation and Injection in Mandarin-English ASR.