Adapting ASR Models for Speech-to-Punctuated-Text Recognition with Utterance Gluing

Agata Jakubiak, Piotr Stachyra, Piotr Czubowski, Hubert Borkowski, Sebastian Łątka,
Radosław Iżak, **Kornel Jankowski**, **Sonia Janicka** and **Mateusz Zieliński**
Samsung R&D Institute Poland
{a.jakubiak2,p.stachyra,h.borkowski,s.latka}@partner.samsung.com,
{p.czubowski,r.izak,k.jankowski,s.janicka,m.zielinski3}@samsung.com

Abstract

Punctuation prediction is a necessary part of ASR models, usually accomplished in a cascaded framework, where a secondary text-based model supplements an unpunctuated ASR output with punctuation marks. However, this approach results in ignoring acoustic context, which makes it poorly suited to certain languages. In this paper, we explore previously proposed ideas on an alternative approach, i.e. Speech-To-Punctuated-Text (STPT) models, and present a solution that allows adapting existing ASR models to output punctuated text. Additionally, we propose utterance gluing, a method of augmenting data to circumvent the lack of speech corpora with long utterances and punctuated references. Our STPT models trained on augmented data outperform STPT models trained on regular data, as well as traditional cascaded models, suggesting that acoustic-based punctuation prediction may be a good alternative to the more common text-based punctuation prediction.

1 Introduction

With the advances in Automatic Speech Recognition (ASR), speech recognition models have become useful in many contexts. Still, there are areas in ASR research which, despite their influence on practical usage, remain under-researched. One of these is punctuation prediction – the task of giving proper punctuation to the ASR output.

Appropriate punctuation in a text is important both for its readability to humans (Ákos Tündik et al., 2018), and for the success of downstream tasks which use it as input, such as machine translation (Vandeghinste et al., 2018) or named entity recognition (Nguyen et al., 2020). Long blocks of text, if not separated into sentences, can be difficult for humans and machines to parse through and understand; additionally, some sentences may be ambiguous without appropriate punctuation. For these reasons, no matter the use-case of an ASR model, having a properly punctuated output is generally preferable.

Despite this, a still widely-used approach to ASR models is to make them output unpunctuated, lowercase text. Such text is often subject to a separate process called punctuation prediction (Gravano et al., 2009), which adds punctuation to it. Many punctuation prediction models do not use any acoustic features present in speech, relying only on the text output of ASR as their input; this is referred to as lexical punctuation prediction. However, this approach presents issues.

Firstly, if a text may be correctly punctuated in multiple ways, it is impossible for the model to distinguish between them without access to acoustic context. This is especially striking in languages that rely more heavily on the acoustic context rather than the grammatical structure of the sentence to disambiguate between different meanings, such as Spanish (Hualde, 2005) or French (Price, 2005), wherein questions are often distinguished from declarative statements exclusively through prosody.

Secondly, since the lexical punctuation prediction relies on the text output of the ASR, any ASR errors are likely to result in punctuation errors, as the punctuation prediction model tries to punctuate the incorrect sentence.

Thirdly, this approach adds the burden of maintaining an additional model alongside the ASR model itself. This is additionally problematic when working with limited memory and computational power, such as when running on mobile devices.

A practiced solution to the *first* and *second* issue is creating hybrid punctuation prediction models which use acoustic features as input alongside text (Klejch et al., 2017), and have access to additional acoustic context not present in the text itself. These models are usually bigger and more complex than purely lexical models, which makes the third issue even more prevalent. A less common solution, which addresses all three issues, is creating

ASR models that directly output punctuated text, and learn to place punctuation marks based solely on the speaker's prosody (Nozaki et al., 2022; Kim et al., 2023). This is referred to as acoustic punctuation prediction, and is the solution we are developing.

The biggest roadblock in developing robust Speech-To-Punctuated-Text (STPT) models is the lack of appropriate speech corpora with both punctuated references and long utterances. Discarding corpora without punctuation marks (e.g., LibriSpeech (Panayotov et al., 2015) and Multilingual LibriSpeech(Pratap et al., 2020)) means severely limiting training data, which unavoidably results in worse recognition metrics, especially in low-resource languages. Moreover, many widely-used speech corpora used for ASR training contain mostly one-sentence utterances (e.g., Common Voice (Ardila et al., 2019)). An STPT model trained on such a dataset is likely to learn to output periods and question marks at the ends of utterances only. This is usually not preferable, as most ASR models are unlikely to process only one sentence at a time.

In this paper, we propose a method of training an STPT model aimed at tackling both these issues without compromising on the Word Error Rate (WER) of the model.

2 Related Work

Creating an end-to-end ASR model that takes speech as input and outputs punctuated text has been previously undertaken for English and Japanese (Nozaki et al., 2022) and for English (Kim et al., 2023).

Mimura et al. (2021) tackled a close topic; however, their goals were much broader, including removal of filler words and changing the speech to be more formal, so their findings are largely inapplicable to our research.

Recently, STPT models have become much more popular, with models such as NVIDIA's Parakeet[1] and Canary[2] being published. These projects did not focus on punctuation; they used punctuated and capitalized transcripts as the training data, so the models learned to produce punctuation in the output, but the creators do not claim to have used any specific methods to improve punctuation results, and they do not share any metrics showing their punctuation performance. We will be focusing on the punctuation-oriented research of Nozaki et al. (2022) and Kim et al. (2023) in our analysis.

2.1 Architecture changes

The main innovation suggested by Nozaki et al. (2022) on creating an STPT model is the addition of an auxiliary loss in an intermediate layer. In their experiments, this addition improved the performance of the model in multiple metrics; however, in the experiments conducted by Kim et al. (2023), the auxiliary loss did not seem to improve the performance of the model significantly.

Kim et al. (2023) focused on streaming, chunk-based ASR, in which their model was only provided with fragments of sentences at a time. This, as explored in more detail in Section 2.2, seems to make punctuation detection much more difficult.

2.2 Punctuation in long utterances

Nozaki et al. (2022) acknowledge that the English training corpus they use, MuST-C (Di Gangi et al., 2019), contains only single-sentence utterances, but they do not attempt to solve this issue. Their model achieves good results on single-sentence test cases, but they do not test it on longer utterances. Their Japanese test utterances are single-sentence only, while only one-sixth of the training ones contain more than one sentence.

Kim et al. (2023) also used MuST-C, but addressed the problem in two ways. Firstly, they concatenated random pairs of training utterances, so that every new utterance consisted of two sentences. Additionally, they also tested the model on long-form speech. The results on long-form test cases were worse than those achieved by Nozaki et al. (2022) on single-sentence test cases, particularly on periods and question marks. However, the model presented by Kim et al. (2023) achieved worse results on periods in single-sentence test cases than it did on periods in long-form test cases, which counter-intuitively suggests that it was actually better at predicting mid-utterance periods than it was at predicting utterance-ending ones. This is likely caused by the fact that its streaming ASR had access to less context, which made it difficult for the model to detect ends of utterances.

[1] https://huggingface.co/nvidia/parakeet-tdt-0.6b-v2

[2] https://huggingface.co/nvidia/canary-qwen-2.5b

3 Proposed Method

Broadly speaking, we wanted our method to be as easy to adapt and use as possible. Because of that, the ideas we propose are focused on data processing, and could be implemented to add punctuation prediction to any ASR model; although, as mentioned before in relation to (Kim et al., 2023), some architectures seem better suited to the task of punctuation prediction than others.

3.1 Punctuation adaptation

In our research, we decided to adapt regular ASR models on punctuated data, rather than training STPT models from scratch. This has many advantages; namely, adapting a model for punctuation prediction is much faster and less resource-intensive than training an STPT model, which is practical for production contexts where time needed to deploy a new model is a factor. Additionally, with this method, training corpora without proper punctuation can still be used in the early phases of training to improve the final ASR model. Finally, with punctuation adaptation, anyone can add punctuation prediction to their existing ASR model, without restarting the training process, which makes the method easier to test and use.

3.2 Utterance gluing

As previously described, since many ASR corpora contain only one sentence in each utterance, STPT models trained on them struggle with placing periods and question marks in places other than the ends of utterances. Concatenating pairs of utterances has been proposed as a solution (Kim et al., 2023); however, an STPT model trained on concatenated utterances could learn to recognize artifacts generated by concatenation (e.g., changes of speakers, loudness, or in the background noise), and place punctuation there. We expanded on the idea of concatenation to make the final utterances resemble natural long-form speech in the following ways:

- Only utterances recorded by the same speaker are concatenated.

- Utterances shorter than 1 second and very quiet utterances (with RMS amplitude lower than 0.01) are discarded.

- Every speaker's utterances are sorted by RMS amplitude, and concatenated with the ones next to them on the sorted list, so that the concatenated utterances have similar volumes.

- Groups of variable numbers of utterances are concatenated, so that the model does not learn to rely on the number of sentences in an utterance.

- A short cross-fade (randomly chosen between 8, 10 and 12 ms) is added between the utterances.

- Long periods of silence from the resulting utterance are cut out, by randomly choosing $duration$ between 0.6, 0.7, 0.8 and 0.9 seconds, and cutting out all parts of the recording that are quieter than 0.2% of the maximum amplitude of a given recording and longer than $duration$. A fragment of silence n seconds long (where n is a random length shorter than $duration$) is left behind, so that some silence remains.

We call this method *utterance gluing*, as it is more complex than simple concatenation. The script used can be found online[3].

3.3 Data processing

We decided to support recognizing periods, commas, question marks, inverted question marks (¿), exclamation marks, and inverted exclamation marks (¡). Our data processing pipeline for punctuation data was as follows:

1. All punctuation marks other than those supported were removed from the reference text. Additionally, all periods used in abbreviations and initials were removed.

2. Every occurrence of a supported punctuation mark was replaced by a tag, written as a separate word; those tags were also placed in the token vocabulary of the model.

4 Models

4.1 ASR

The ASR model used in this work is a conformer-transducer, a sequence-to-sequence model, which is a variation of an architecture derived from the RNN-transducer (Graves, 2012). Specifically, we

[3] https://github.com/SamsungLabs/adapting-asr-models-for-stpt-with-utterance-gluing

employ the first-pass model architecture as described in section 2 in (Park et al., 2023) without a feedback path from the joiner to the predictor. We refrained from using the second-pass portion of the architecture, focusing on the applicability of the proposed method to a single-pass streaming model. We release the code used for training on GitHub[3].

The concept relies on employing transcriber and predictor networks: the former operating on the acoustic features $\mathcal{X} \in \mathbb{R}^d$ derived from the audio signal, the latter on the utterance transcription encoding $\mathcal{Y}$, representing wordpieces.

The transcriber takes an input sequence of acoustic features and outputs a transcription vector. In this work, the transcriber is a stack of 16 conformer layers (Gulati et al., 2020) capturing the global, as well as local patterns by utilizing attention and convolution layers. To ensure optimal resources utilization, we used striding as a reduction technique applied to the acoustic features, prior to processing by the transcriber.

The predictor consists of two layers of an LSTM network. Its purpose is to learn to model an output sequence $g = (g_0, g_1, ..., g_U)$, where U corresponds to the tokens' sequence length.

It is worth noting that the input sequence is the original tokens' sequence $y = (y_1, ..., y_U)$ with an encoded null output $\emptyset$, prepended to it. Therefore, at the input, we process an extended input vector $\hat{y} = (\emptyset, y_1, ..., y_U)$, as proposed by previous work (Graves, 2012). Utilization of a blank token enables teaching the model how to align speech, i.e. account for silent parts in utterances without malforming the transcribed speech sequence in temporal context.

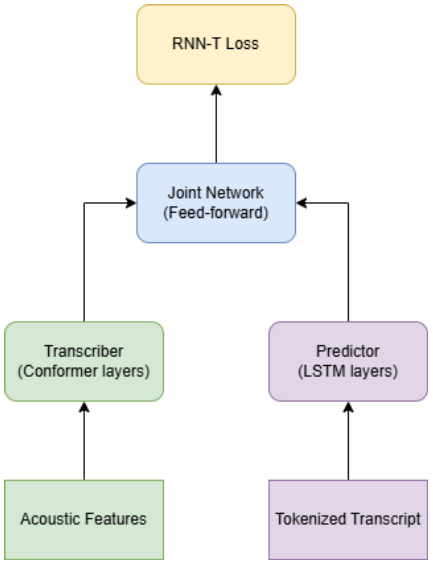
These networks are jointly trained using a Joiner, integrating the information from both networks, with an objective function (commonly known as RNN-T Loss) defined as log posterior probability: $\mathcal{L} = -ln(y|x)$. Joiner adds the outputs of transcriber and predictor, which are further passed through activation layer and linear layers.

The ASR we trained had 30 million parameters. An overview of the architecture used for the ASR model used in this work is shown in Figure 1.

4.2 Lexical restoration

To evaluate our approach against lexical methods, we also trained and tested transformer-based token classification models. This was done due to the lack of appropriate open-source models for this study; the most appropriate being KREDOR's punctuate-all model[4], based on (Guhr et al., 2021), which does not support exclamation marks and inverted punctuation marks. For each language, an instance of XLM-RoBERTa-large (Conneau et al., 2019) was first fine-tuned on a mix of long- and short-form utterances with a 1:4 ratio, and then further trained on the former only. The needed datasets were accessed through the OPUS (Tiedemann, 2012) website and included ParaCrawl (Bañón et al., 2020), OpenSubtitles (Lison and Tiedemann, 2016), and EuroParl (Koehn, 2005) to balance formal and informal writing styles. For each dataset, short-form sentences were retrieved and cleaned (e.g., abbreviations were removed). Then, a random subsample was concatenated to form utterances 2-6 sentences long. In total, each model was trained on more than 16 M utterances per epoch, with training ending after 15 epochs, or if the average of all punctuation mark metrics plateaued for more than two epochs.

Figure 1: Transducer architecture used in this work.



5 Experiments

5.1 Datasets used

We decided to run our experiments on German, Polish, and Spanish, as those languages represent three different language subgroups (Eberhard et al., 2024), and we suspected that different approaches to punctuation prediction might work best for different kinds of languages. Unfortunately, we could not train an English model with MuST-C and compare it to previous works on this subject, (Nozaki

[4] https://huggingface.co/kredor/punctuate-all

et al., 2022) and (Kim et al., 2023), since the dataset is not currently available[5].

5.1.1 Training and validation datasets

For punctuation training purposes, we searched for open-source datasets with well-punctuated references. We decided to use Common Voice 16.1 (Ardila et al., 2019) for Spanish and German, and Common Voice 13.0 with ParlaSpeech (Ljubešić et al., 2025) for Polish. 1% of the data was selected for validation. The number of utterances and punctuation marks in each dataset can be seen in Table 1.

For the purposes of our experiments, we created four versions of each training and validation dataset:

1. A non-glued, non-punctuated version, used to train a regular ASR model.

2. A non-glued, punctuated version, with most of the utterances only containing one sentence, later referred to as "single-sentence punctuated data" (*single*).

3. A concatenated, punctuated version, where utterances were randomly concatenated into groups of 2-3, resulting in 361k utterances in German, 230k in Polish and 591k in Spanish, and their references concatenated accordingly (*concat*).

4. A glued, punctuated version, where utterances were glued together into groups of 2-3, using the methodology described in section 3.2, resulting in 339k utterances in German, 199k in Polish and 549k in Spanish, and their references concatenated accordingly (*glued*).

Table 1: Number of utterances and punctuation marks in original non-augmented datasets.

Language	**Utts**	**.**	**,**	**¿**	**?**	**¡**	**!**
German	867k	801k	218k	0	47k	0	22k
Polish	556k	446k	578k	0	51k	0	69k
Spanish	1418k	1418k	508k	5.7k	5.7k	4.5k	8.8k[6]

5.1.2 Evaluation datasets

We needed to use real multi-sentence utterances to evaluate the models on actual mid-utterance periods, question marks and exclamation marks. We decided to use Multilingual LibriSpeech (MLS), which contains many long utterances from audiobooks (Pratap et al., 2020). The released version of this dataset does not contain punctuation in its references, but we restored the punctuation using the original books' text. Then, for each language, we selected 1024 utterances which contained at least one question mark from the training subset of the corpus, and we manually modified the references to only contain the punctuation marks we were using (e.g., replacing semicolons with periods). We did not simply remove the unsupported punctuation marks, as we did in training data, because MLS contained much more of them than our training datasets. However, we removed a few utterances which contained punctuation that could not be straightforwardly replaced. The dataset details can be seen in Table 2. The evaluation datasets were released on GitHub[3].

Table 2: Number of punctuation marks in evaluation datasets.

Language	**Utts**	**.**	**,**	**¿**	**?**	**¡**	**!**
German	1020	1825	3210	0	1421	0	429
Polish	1014	2958	4051	0	1364	0	351
Spanish	1022	2525	3134	1338	1338	323	323

5.2 Experiment methodology

In our experiment, we wanted to compare the effectiveness of the following approaches: *lexical* restoration and three variants of acoustic recognition: trained on *single*, *concat*, and *glued* punctuated data.

5.2.1 Acoustic model training

To that end, firstly, we trained a multilingual ASR model from scratch for 925k steps on the non-punctuated version of all three training datasets. Then, we adapted it on the non-punctuated training dataset for every language, resulting in three regular, non-punctuated ASR models. Then we adapted each of them on the *single*, *concat*, and *glued* punctuated data, resulting in three different STPT models for every language. Table 3 shows the numbers of training steps for each checkpoint chosen for evaluation.

[5] `https://mt.fbk.eu/resources/` accessed 2025-01-21

[6] Although Spanish Common Voice has an unequal number of opening and closing exclamation marks, and very few question marks, it was still the best dataset available for our purpose.

5.2.2 Vocabulary

The token vocabulary of all of the models was the same. Tags used for punctuation prediction were present in the vocabulary from the start, and went unused by the earlier, non-punctuated models. Therefore, adaptations consisted simply of running training from a previously trained checkpoint, with entirely new training and validation data, and no other changes. When adapting a previously trained ASR model with no punctuation tags in the vocabulary, one could accomplish the same outcome by replacing the least used tokens in the vocabulary with punctuation tags. This would allow the model to adapt for punctuation prediction without the size of the vocabulary being changed, and without the need to retrain the model from scratch.

5.2.3 Lexical models

Additionally, for every language, we used our lexical punctuation prediction model (as described in Section 4.2) and KREDOR's punctuate-all model to create two cascaded, lexical STPT models out of the non-punctuated ASR models created in 5.2.1, in order to compare the acoustic models with state-of-the-art lexical punctuation prediction. It is worth mentioning that our lexical models are more than 18 times larger, and KREDOR is about 9 times larger, than our STPT models.

5.2.4 Performance metrics

To compare these approaches, we treated them as if the models were binary classifiers deciding whether or not the given punctuation mark should be placed at a given position in the recognized text and compared their precision, recall, and F1 scores. Additionally, we compared WERs of the models with punctuation marks excluded.

Table 3: Number of training steps for chosen checkpoints.

Language	non-punct	single	concat	glued
German	1891k	2143k	2000k	1980k
Polish	1569k	1703k	1600k	1654k
Spanish	1960k	2420k	2140k	2155k

5.3 Results and discussion

The evaluation results of the five previously described approaches for each language can be seen in Table 4. Since the lexical models used the outputs of non-punctuated ASR models, the WERs listed in the lexical models' rows are the WERs of acoustic models before punctuation adaptations. They can be also used to see how punctuation adaptations affected WERs.

5.3.1 Exclamation marks

In our experiments, exclamation marks could not be reliably recognized by any model (best F1 score was 0.21, and most were far worse). In acoustic models, this does not seem to stem from them being underrepresented in training data (see Table 1). It is likely they are close enough to periods, both in their pronunciation and their usage, that neither lexical nor acoustic model can tell them apart. Since mistaking exclamation marks for periods does not usually impact the meaning of the text, we decided to treat exclamation marks as equivalent to periods in our results, and disregard inverted exclamation marks.

5.3.2 Lexical models

Our *lexical* models achieved similar results to KREDOR's state-of-the-art model, with the notable exception of question marks, where their results were better. For that reason, going forward, we will be using them as the lexical state-of-the-art benchmark. Although our models were trained on very similar data to each other, some metrics differ strongly between languages. This suggests that lexical punctuation prediction may be better suited for some languages than for others.

5.3.3 Acoustic models

In general, the *single* acoustic models performed very poorly, achieving the lowest F1 scores out of the acoustic models on all languages and punctuation marks, except for Spanish utterance-ending periods. As predicted, they were almost unable to produce mid-utterance periods and question marks, with the notable exception of Spanish mid-utterance periods.

In Polish and German, the *glued* models achieved the highest F1 scores on all punctuation marks, outperforming all other models, both acoustic and lexical. The most notable difference between *lexical* and *glued* models was in mid-utterance periods and mid-utterance question marks, though in Polish the difference on utterance-ending question marks was also large.

In Spanish, there is no clear best-performing model. Our Spanish acoustic models were by far the worst of the three languages at recognizing question marks, and they were outperformed by the *lexical* model. This is likely caused by question

Table 4: Comparison of recalls, precisions and F1 scores of punctuation marks' recognition between models. For sentence-ending punctuation marks, results are split into mid-utterance and utterance-ending marks. Exclamation marks have been treated as periods, and inverted exclamation marks have been deleted. WER values are calculated with punctuation marks excluded.

Language	Model	WER	mid .			end .			,			mid ?			end ?			¿		
			Rec	Pre	F1	Rec	Pre	F1	Rec	Pre	F1	Rec	Pre	F1	Rec	Pre	F1	Rec	Pre	F1
German	KREDOR	0.24	0.49	0.59	0.53	0.91	0.69	0.79	**0.65**	0.65	**0.65**	0.32	0.67	0.44	0.48	0.83	0.61	-	-	-
	lexical	0.24	0.50	0.53	0.52	0.88	**0.72**	0.79	0.64	0.67	**0.65**	0.41	0.62	0.49	**0.53**	**0.88**	**0.66**	-	-	-
	single	0.21	0.02	**0.90**	0.03	0.90	0.64	0.75	0.64	0.51	0.57	0.01	0.73	0.02	0.37	0.76	0.49	-	-	-
	concat	**0.19**	0.57	0.72	0.63	**0.93**	0.69	0.80	0.63	0.66	0.64	0.34	**0.77**	0.47	0.48	0.86	0.61	-	-	-
	glued	**0.19**	**0.70**	0.67	**0.68**	**0.93**	0.71	**0.81**	0.62	**0.68**	**0.65**	**0.49**	0.76	**0.59**	**0.53**	0.86	**0.66**	-	-	-
Polish	KREDOR	0.28	0.46	0.59	0.52	0.95	0.78	0.86	0.63	0.63	0.63	0.28	0.66	0.39	0.46	0.83	0.60	-	-	-
	lexical	0.28	0.46	0.57	0.51	0.94	0.78	0.85	0.61	0.62	0.62	0.39	0.61	0.47	0.48	0.85	0.61	-	-	-
	single	0.24	0.00	0.32	0.01	0.92	0.73	0.81	0.67	0.49	0.56	0.05	**0.89**	0.09	0.32	0.67	0.44	-	-	-
	concat	0.22	0.32	**0.78**	0.45	0.94	0.79	0.86	**0.68**	0.57	0.62	0.46	0.85	0.60	0.50	0.81	0.62	-	-	-
	glued	**0.21**	**0.50**	**0.78**	**0.61**	**0.96**	**0.85**	**0.90**	0.61	**0.67**	**0.64**	**0.67**	0.82	**0.74**	**0.66**	**0.88**	**0.76**	-	-	-
Spanish	KREDOR	0.24	0.39	0.55	0.45	0.99	0.54	0.70	**0.52**	0.47	0.50	0.06	**0.59**	0.11	0.07	0.88	0.14	-	-	-
	lexical	0.24	0.45	0.54	0.49	0.91	0.63	0.74	0.44	0.52	0.48	**0.24**	0.54	**0.33**	**0.34**	0.87	**0.49**	**0.31**	0.73	**0.43**
	single	0.33	0.27	0.68	0.39	**0.99**	**0.64**	**0.78**	0.36	0.50	0.42	0.02	0.41	0.03	0.01	0.75	0.02	0.03	0.62	0.05
	concat	0.17	0.52	**0.76**	0.62	0.97	0.58	0.72	0.51	0.54	**0.53**	0.20	0.44	0.28	0.21	0.87	0.34	0.25	0.62	0.36
	glued	**0.16**	**0.74**	0.63	**0.68**	0.98	0.56	0.71	0.40	**0.60**	0.48	0.22	0.55	0.32	0.14	**0.88**	0.24	0.24	**0.75**	0.36

marks being underrepresented in the Spanish training corpus. In internal experiments which utilized glued non-public data of better balance, higher results were achieved (0.39 recall and 0.88 precision for mid-utterance question marks, 0.38 recall and 0.94 precision for utterance-ending question marks, 0.35 recall and 0.88 precision for inverted question marks; for other punctuation marks, the results were comparable to the *glued* model).

5.3.4 Effects on WER

The WER seems positively affected by concatenation and gluing, although all acoustic models had access to the same training data, just processed differently. We think this is linked to the fact that the evaluation data consists of long utterances; it seems that training ASR models on long utterances improves their performance in recognizing long utterances.

5.3.5 Checkpoint instability

It is important to mention that during our training runs, the punctuation results between even close checkpoints varied strongly; it seemed difficult for an STPT model to find a local minimum for a punctuation task, as the model was trained for minimizing WER in general, without any special optimization for punctuation. It is likely that a training method with two loss functions, one aimed at minimizing WER and the other at optimizing the punctuation performance, could be used to improved the results further. That being said, we have trained our models for a significant time, and the checkpoints we are presenting are the best of many, so we are reasonably sure that these are the best punctuation results possible with this method, despite the variability.

5.4 Possible new issues

We have found that acoustic punctuation prediction addresses issues inherent to lexical punctuation prediction, namely lexical ambiguity and dependence on good ASR output for good results. In our hands-on experiments, for example, a strong questioning tone of voice was enough to produce a question mark, regardless of whether the phrase spoken was grammatically a question, a statement, or even incoherent babble.

However, this approach creates new issues that need to be discussed. Some speakers may have a flat tone of voice that does not indicate a question when they are asking one. Some may pause while speaking, without intending for a comma or a period to be placed. In general, the performance of acoustic punctuation prediction is more dependent on the speaker, and how clearly they are speaking, and less dependent on whether the phrases they are using are grammatically correct, and have been recognized correctly.

Since we have proven that acoustic models can outperform lexical models, it seems that these issues are less prevalent than the ones present in lexical models, at least in our test cases.

6 Conclusions

In this paper, we postulate that acoustic punctuation prediction is a strong alternative to lexical punctuation prediction. We show that multi-sentence training utterances are necessary for training well-functioning STPT models, and that punctuated training corpora with single-sentence utterances can be augmented to be used for STPT model training. We theorize about the problems caused by concatenation, and we address them by developing our gluing technique. We show that gluing improves the results over concatenation (weighted avg F1 equal 0.5725 and 0.5371, respectively), and that both methods are superior to training acoustic models on single-sentence utterances. We also show that acoustic models can outperform lexical punctuation prediction models (with weighted avg F1 equal 0.4857), despite being much smaller.

7 Future work

The biggest challenge of end-to-end STPT models is the lack of well-punctuated corpora with multi-sentence utterances. This work was an attempt to circumvent that, and could be developed by improving the gluing methods further; however, if real long-utterance corpora were developed, the models trained on them would likely outperform the ones presented here, and possibly any model trained on glued data. Additionally, as we showed that languages can be better or worse suited for different approaches to punctuation prediction, we hope that more research on the topic will be conducted with non-English languages in mind.

Since the acoustic punctuation prediction is gaining popularity, as seen in models such as NVIDIA's Parakeet[1] and Canary[2], we believe it is important to measure and share the punctuation results of STPT models and work to improve these results, instead of treating punctuation as an afterthought. Judging by the high-quality outputs of these models, even though the authors did not share punctuation metrics, it seems that English STPT models can be trained on non-augmented punctuated data from scratch, since there is quite a large amount of such English data. For other languages, methods presented in this paper may be needed.

Lastly, we suggest that future efforts in developing speech corpora include punctuation in their references if possible, to enable further developments in this field.

8 Limitations

In our work, we have shown the advantage of acoustic models over lexical models when it comes to small ASR models trained on relatively small corpora, with relatively high WER. However, high WER negatively impacts the performance of lexical models, as the input they receive is unreliable. It would be useful to test these methods on larger, better-performing ASR models, and find if acoustic models continue to outperform lexical ones when the WER is lower.

Additionally, we have focused on one specific architecture – the sequence transducer – in our work. We hope the methods shown here are transferrable to different architectures, as none of our methods were reliant on the features of the sequence transducer. However, it is possible that different architectures differ in their suitability for use for STPT, and we do not know if the results shown here are representative of how every architecture would perform. This has to be investigated further to reach any definite conclusions.

9 Acknowledgements

GNU Parallel (Tange, 2011) was used in our processing pipelines. We recognize the efforts of the developers of the Torch library (Paszke et al., 2017), with a strong emphasis on its audio-oriented components (Hwang et al., 2023), as well as the Pytorch-Lightning framework[7], as essential for the democratization of artificial intelligence research, which was vital for this work.

References

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. 2019. Common voice: A massively-multilingual speech corpus. *ArXiv*, abs/1912.06670.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

[7] https://www.pytorchlightning.ai

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv*, abs/1911.02116.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.

David M Eberhard, Gary F Simons, and Charles D Fennig. 2024. Ethnologue: Languages of the World. Twenty-seventh edition. Dallas, Texas: SIL international.

Agustin Gravano, Martin Jansche, and Michiel Bacchiani. 2009. Restoring punctuation and capitalization in transcribed speech. *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4741–4744.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *ArXiv*, abs/1211.3711.

Oliver Guhr, Anne-Kathrin Schumann, Frank Bahrmann, and Hans Joachim Böhme. 2021. Fullstop: Multilingual deep models for punctuation prediction. In *Proceedings of the Swiss Text Analytics Conference 2021*, Winterthur, Switzerland. CEUR Workshop Proceedings.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition.

José Ignacio Hualde. 2005. *The sounds of Spanish*. Cambridge University Press.

Jeff Hwang, Moto Hira, Caroline Chen, Xiaohui Zhang, Zhaoheng Ni, Guangzhi Sun, Pingchuan Ma, Ruizhe Huang, Vineel Pratap, Yuekai Zhang, Anurag Kumar, Chin-Yun Yu, Chuang Zhu, Chunxi Liu, Jacob Kahn, Mirco Ravanelli, Peng Sun, Shinji Watanabe, Yangyang Shi, and Yumeng Tao. 2023. TorchAudio 2.1: Advancing speech recognition, self-supervised learning, and audio processing components for Pytorch. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–9.

Hanbyul Kim, Seunghyun Seo, Lukas Lee, and Seolki Baek. 2023. Improved training for end-to-end streaming automatic speech recognition model with punctuation. In *Interspeech 2023*, pages 1653–1657.

Ondřej Klejch, Peter Bell, and Steve Renals. 2017. Sequence-to-sequence models for punctuated transcription combining lexical and acoustic features. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5700–5704.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Nikola Ljubešić, Peter Rupnik, and Danijel Koržinek. 2025. The ParlaSpeech collection of automatically generated speech and text datasets from parliamentary proceedings. In *Speech and Computer*, pages 137–150, Cham. Springer Nature Switzerland.

Masato Mimura, Shinsuke Sakai, and Tatsuya Kawahara. 2021. An end-to-end model from speech to clean transcript for parliamentary meetings. In *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 465–470.

Thai Binh Nguyen, Quang Minh Nguyen, Thi Thu Hien Nguyen, Quoc Truong Do, and Chi Mai Luong. 2020. Improving vietnamese named entity recognition from speech using word capitalization and punctuation recovery models. In *Interspeech 2020*, pages 4263–4267.

Jumon Nozaki, Tatsuya Kawahara, Kenkichi Ishizuka, and Taiichi Hashimoto. 2022. End-to-end speech-to-punctuated-text recognition. In *Interspeech 2022*, pages 1811–1815.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an ASR corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Jinhwan Park, Sichen Jin, Junmo Park, Sungsoo Kim, Dhairya Sandhyana, Changheon Lee, Myoungji Han, Jungin Lee, Seokyeong Jung, Changwoo Han, and Chanwoo Kim. 2023. Conformer-based on-device streaming speech recognition with kd compression and two-pass architecture. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 92–99.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS-W*.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. 2020. MLS: A large-scale multilingual dataset for speech research. In *Interspeech 2020*, pages 2757–2761.

Glanville Price. 2005. *Intonation*, chapter 20. John Wiley & Sons, Ltd.

Ole Tange. 2011. Gnu parallel - the command-line power tool. *login: The USENIX Magazine*, February 2011:42–47.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Vincent Vandeghinste, Lyan Verwimp, Joris Pelemans, and Patrick Wambacq. 2018. A comparison of different punctuation prediction approaches in a translation context. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 289–298, Alicante, Spain.

Máté Ákos Tündik, György Szaszák, Gábor Gosztolya, and András Beke. 2018. User-centric evaluation of automatic punctuation in asr closed captioning. In *Interspeech 2018*, pages 2628–2632.