Hyper-BTS Dataset: Scalability and Enhanced Analysis of Back TranScription (BTS) for ASR Post-Processing

Chanjun Park[1], **Jaehyung Seo**[2], **Seolhwa Lee**[3,4], **Junyoung Son**[2]
Hyeonseok Moon[2], **Sugyeong Eo**[2], **Chanhee Lee**[5], **Heuiseok Lim**[2,†*]

[1]Upstage AI, [2]Korea University, [3]Technical University of Darmstadt, [4]Linq, [5]Naver Corporation
chanjun.park@upstage.ai
{seojae777, s0ny, glee889, djtnrud, limhseok}@korea.ac.kr
chanhee.lee@navercorp.com

Abstract

The recent advancements in the realm of Automatic Speech Recognition (ASR) post-processing have been primarily driven by sequence-to-sequence paradigms. Despite their effectiveness, these methods often demand substantial amounts of data, necessitating the expensive recruitment of phonetic transcription experts to rectify the erroneous outputs of ASR systems, thereby creating the desired training data. Back TranScription (BTS) alleviates this issue by generating ASR inputs from clean text via a Text-to-Speech (TTS) system. While initial studies on BTS exhibited promise, they were constrained by a limited dataset of just 200,000 sentence pairs, leaving the scalability of this method in question. In this study, we delve into the potential scalability of BTS. We introduce the "Hyper-BTS" dataset, a corpus approximately five times larger than that utilized in prior research. Additionally, we present innovative criteria for categorizing error types within ASR post-processing. This not only facilitates a more comprehensive qualitative analysis, which was absent in preceding studies, but also enhances the understanding of ASR error patterns. Our empirical results, both quantitative and qualitative, suggest that the enlarged scale of the Hyper-BTS dataset sufficiently addresses a vast majority of the ASR error categories. We make the Hyper-BTS dataset publicly available.[1]

1 Introduction

A large-scale dataset-based NLP research paradigm, which is based on foundation models (Bommasani et al., 2021) such as GPT-4 (OpenAI, 2023), and prompt tuning using natural-language prompts (Liu et al., 2021) has recently been of interest in both the academia and industry. Such large-scale models have proven that there is efficiency in the usage of large-scale datasets, and include a scaling law model (Kaplan et al., 2020), which theoretically demonstrates their justification.

There is also increasing application of this promising research paradigm in the Automatic Speech Recognition (ASR) field. Aside from traditional speech recognition architecture-based research such as Gaussian Mixture Models (GMMs) (Stuttle, 2003), and Hidden Markov Models (HMMs) (Gales and Young, 2008) based on acoustic and language models, model-centric ASR research using transfer learning based on pre-trained models is currently being widely conducted (Baevski et al., 2020; Giollo et al., 2020; Hjortnæs et al., 2021; Zhang et al., 2021).

Model-centric ASR research requires the configuring of many parameters for the pre-training of models, as well as a sufficiency of computing power (*e.g.,* GPU) to process large-scale datasets. Thus, despite its proven efficiency, insufficiency of computing power in real-world service scenarios limits the performance of this ASR model approach. In other words, since many parameters and data are required when training a model, companies that do not have sufficient server or GPU environments have difficulty configuring service environments and improving performance using the model-centric ASR approach (Park et al., 2020b).

Conversely, a different approach, termed "data-centric" has also emerged, which aims to improve ASR model performance by improving the data quality or pre-processing and post-processing without model modification (Voll et al., 2008; Mani et al., 2020; Liao et al., 2020; Park et al., 2021a). This alleviates the previous limitations (of computation cost and non-scalable human annotation) because it does not modify the model, and enables its application to lightweight models such as the vanilla Transformer, which can be sufficiently processed by a single CPU (Vaswani et al., 2017; Klein et al., 2020).

[†] Corresponding author

[1] https://github.com/Parkchanjun/HyperBTS

Findings of the Association for Computational Linguistics: EACL 2024, pages 67–78
March 17-22, 2024 ©2024 Association for Computational Linguistics

There has been a recent endeavor in the data-centric ASR post-processor approach known as Back Transcription (BTS) (Park et al., 2021b). BTS, an automatic data construction method, has been devised for use as substitute for publicly available training data, for ASR post-processor based on a sequence-to-sequence model (converting input sequences into target sequences) and to eliminate the requirement to build parallel corpora by human-annotators. Specifically, this method integrates Text-to-Speech (TTS) with Speech-to-Text (STT) efficiently for building a pseudo-parallel corpus (see detail in Appendix A).

However, in a current BTS study, model training was performed using only a 200,000 parallel corpus in Korean. While this may be a significant amount from the point of view of low-resource Neural Machine Translation (NMT), it is very small in comparison with the recent research flow utilizing large-size data. In addition, only the method and demo system were disclosed in the BTS study, but no dataset was released with the work. Therefore, to improve the performance of an ASR post-processor, based on BTS technology, we take advantage of the existing research flow to build large-capacity data and present a Hyper-BTS dataset that is five times the size of the existing BTS study, with a one million-text large-capacity dataset. Further, to activate the relevant research interest, we make it publicly accessible, dividing the data into training, validation, and test datasets. To the best of our knowledge, this is the first time a parallel corpus for an ASR post-processor has been made public. By opening the data in this way, ASR post-processor research can be triggered, and the problems with the existing commercial ASR API systems can be studied and improved.

Existing commercial ASR APIs currently present problems such as spacing, conversion of numbers, and pronunciation boundary errors. Therefore, it is inevitable that ASR post-processor recognition results will contain unexpected errors. In other words, there is room for performance improvement using ASR post-processor, and additionally, precise error analysis is required.

Despite the acknowledgement of the existence of recognition errors, there are currently no precise criteria for categorizing output error types from ASR systems. Many studies related to large-scale language models (Baevski et al., 2020; Zhang et al., 2021) have through their works attempted to develop a model (Gales and Young, 2008) for improving ASR systems. However, analysis of the types of errors output by ASR systems and guidelines on research and design are insufficient, as existing studies simply analyze the advantages and disadvantages of generated results without benchmarking the results against some set of standards.

In this study, we propose novel criteria of error type categorization of ASR post-processor specialized in Korean, in terms of BTS work also based on Korean. We present this set of criteria to be used for the direction of further work in enhancing ASR post-processor performance. In addition, based on our defined error types, we perform an in-depth qualitative analysis of the Hyper-BTS dataset-based ASR post-processor to verify whether actual error correction is performed well. Through this, we suggest methods that can be employed to improve the performance of ASR post-processor systems.

The contributions of this study are as follows:

- We released a large-scale Hyper-BTS dataset, five times larger than the existing BTS dataset, separated into training, validation, and test sets. It is the first published parallel corpus for ASR post-processor to the best of our knowledge.

- Our various quantitative analyses of ASR post-processor experiments using the Hyper-BTS dataset demonstrate an objective performance of the corresponding dataset.

- We proposed a detailed error classification criterion for Korean, which has significantly different linguistic characteristics from other languages, and based on this, we performed a qualitative analysis on the Hyper-BTS dataset-based ASR post-processor to verify the dataset. Our analysis results enable us to present a method that can be used to improve the performance of ASR post-processor systems.

2 Hyper-BTS Dataset

2.1 Dataset Design

Build Mono Corpus As a language pair to construct the Hyper-BTS dataset, we arrange it in the same language as the present BTS paper and gather monolingual corpus from three sources.

Hyper-BTS	Train		Valid		Test	
	src	tgt	src	tgt	src	tgt
# of sents	1,000,000	1,000,000	5,000	5,000	3,000	3,000
# of tokens	32,527,375	34,308,007	140,641	147,390	83,230	87,207
# of words	8,857,758	8,929,016	37,792	37,112	22,388	21,975
avg of SL △	32.66	34.45	28.13	29.48	27.74	29.07
avg of WS	8.89	8.97	7.56	7.42	7.46	7.33
avg of SS	7.89	7.96	6.56	6.42	6.46	6.33
# of K-toks ∗	24,243,741	24,900,124	106,217	107,106	63,077	63,659
# of E-toks	129,281	88,156	517	959	284	509
# of S-toks	13,099	1,282,930	36	6,069	12	3,575

Table 1: Statistics of our Hyper-BTS Dataset. We define the original colloquial sentences as target (tgt) and the generated sentences after BTS as source (src). Moreover, we attempt to identify the linguistic features of our parallel corpus including # of sents/tokens/words: number of sentences/tokens/words; △ avg of SL/WS/SS: average of sentence length/words/spaces per sentence; ∗ # of K-toks/E-toks/S-toks: number of Korean/English/special-symbol letter tokens.

First, 129,987 sentences were excerpted from business and technology TED Talks, provided in writing translated into Korean. Second, 373,013 sentences were discovered, corresponding to the spoken language among Korean-English, and translated parallel corpus from AI-HUB, which is the most reliable and utilized data platform in numerous examinations related to the Korean language. Third, 505,000 sentences were extracted from the National Institute of Korean Language's colloquial corpus.

TTS(Text-To-Speech) The built mono-corpus is converted to voice data in mp3 format, based on the Naver Clova Voice API (Chung, 2019). The 503,000 sentences from TED Talks and AI-HUB were divided into 9,963,296 voice tokens and synthesized into 7,963,935 seconds of voice data. The 505,000 sentences extracted from the spoken corpus of the National Institute of Korean Language were separated into 14,595,647 voice tokens and synthesized into 11,563,990 seconds of voice data. The respective running time was five and six days. The reason for using the commercial system is to lower entry barriers by allowing companies without a built-in TTS system to use BTS.

STT(Speech-To-Text) Naver Clova speech recreation API was used to convert results of TTS voice data to text data. It took 10 and 11 days, respectively, and the total time required was three weeks. The Hyper-BTS dataset of 1,008,000 sentence pairs is eventually established.

The Final Constructed Hyper-BTS Dataset Finally, the Hyper-BTS dataset of 1,008,000 sentences is separated into train-, validation-, and test-sets. Train-set consisted of 1,000,000 sentences, verification-set was 5,000 sentences, and test-set had 3,000 sentences. We attempted to minimize results-to-data sources bias in the test-set by extracting 1,500 sentences from AI-HUB/TED and 1,500 sentences from the colloquial corpus of the National Institute of Korean Language.

2.2 Data Statistics and Analyses

We conducted an in-depth statistical analysis of the Hyper-BTS dataset, as shown in Table 1.

Fundamental Analysis Fundamental analysis was done on the number of sentences, tokens, and average sentence length. First, in the case of sources through Hyper-BTS, the sentence length was shorter by 1.79, 1.35, and 1.33 on average than the original sentence target. The total number of tokens decreased by 5.2%, 4.6%, and 4.6%, respectively. In the case of a target, the number of words in the train-set was 71,258 more than the source. We configured the validation- and test-sets to have different features from the train-set. Therefore, the number of words in the validation and test set decreased by 680,413 from the source in the target, respectively. Considering the average spacing, the total number of words increased even though the total number of tokens was relatively small due to additional unnecessary words in sentences.

Token Analysis in Korean and English The second data statistic is the analysis of Korean and English tokens. The Korean token (K-token) essentially lost 656,383, 889, and 582 train-, validation-, and test-set tokens in source sentences than target sentences, caused by the omission of termination and suffixes. These results reproduce the characteristics of Korean speakers who pronounce endings vaguely in the model. Additionally, the English token (E-token) is transformed into a Korean token as pronounced or omitted because of recognition failure. The train-set lost as much as 41,125 tokens in the target rather than the source. However, we had a significant increase in the number of misaligned transformations from Korean to English, increasing 442 and 225 in the target than in the source for the validation- and test-sets.

Special Token Analysis Third data statistic, special character tokens (S-tokens) show the most notable differences in train-, validation-, and test-sets, as 98.89%, 99.41%, and 99.64% of tokens disappeared from source rather than target sentences. In particular, periods, commas, exclamation marks, and brackets added to describe the situation in the transcription process of the original data have a substantial influence. Such special characters may contain colloquial tones or emotions that the text does not sufficiently represent. Therefore, excessive omission of special characters is like failing to include some of the rich expression information of the spoken language in the written language.

By disclosing the established Hyper-BTS dataset, we attempt to lower the entry barriers of companies and research institutes into the study. This approach can alleviate the cost concerns of many small and medium-sized businesses that do not have individual speech synthesis and recognition technologies.

3 Experiments and Results

3.1 Setting

Experiments Design To determine the effectiveness of the large-scale dataset, we separated the 1 million Hyper-BTS dataset into 10 anchor points. We then trained an ASR post-processor with this corpus and evaluated its performance differences by scaling up the training data size. The experimental results for these are shown in Figure 1.

Next, we adopted parallel corpus filtering (PCF) to the Hyper-BTS dataset, and inspected its impact on the ASR post-processor performance. PCF indicates a selection process that filters out low-quality sentence pairs and acquires high-quality data (Koehn et al., 2020). Particularly in the machine translation (MT) research field, PCF techniques are robustly applied for the performance improvement of MT systems.

Considering the process of constructing the Hyper-BTS dataset, inherent limitations of SST or TTS systems can result in unintended errors. These errors include several outliers such as too short or too long sentences, and omission of the source sentence. We applied the PCF methodologies proposed in Park et al. (2020a) to alleviate these errors and constructed a high quality dataset. The substantial impact of applying the PCF methods can be verified in Table 2.

Finally, we performed a qualitative analysis of a Hyper-BTS dataset-based ASR post-processor in section 3.2. Through investigating post-processor performance results, we propose new ASR post-processor error types and use these to analyze ASR post-processor models. Additionally, we analyzed the practical effectiveness of increasing the size of the Hyper-BTS dataset.

Model Details All the ASR post-processors experimented in this study were built on the transformer-base model structure. These were trained on our Hyper-BTS dataset, and, for the training process, we used the same hyper-parameter setting as Vaswani et al. (2017). For tokenization, we adopted sentence piece (Kudo and Richardson, 2018) model with 32,000 vocabulary size.

Evaluation Details For the evaluation metric, we adopted GLEU (Napoles et al., 2015) and BLEU (Papineni et al., 2002) as in BTS (Park et al., 2021b). GLEU is a correction system specialized metric that is similar to BLEU, but considers source sentences.

3.2 Quantitative Analysis

Importance of Data Size First, we showed the performance improvement that can be obtained by the ASR post-processor, compared with the baseline. In these experiments, the baseline indicates the performance between source sentences and their corresponding target sentences in a test dataset. As shown in Figure 1, compared with baseline whose BLEU score is 40.33, ASR post-processors give significantly surpassing performance for all the anchor points. In particular, ASR post-processor trained with 1 million training data shows 25.31 higher BLEU score over the baseline.

We then inspected the performance difference derived by increasing data size. These are shown in the right side plot of the Figure 1, and denoted "diff". As shown in Figure 1, we can obtain the highest performance by utilizing the whole data

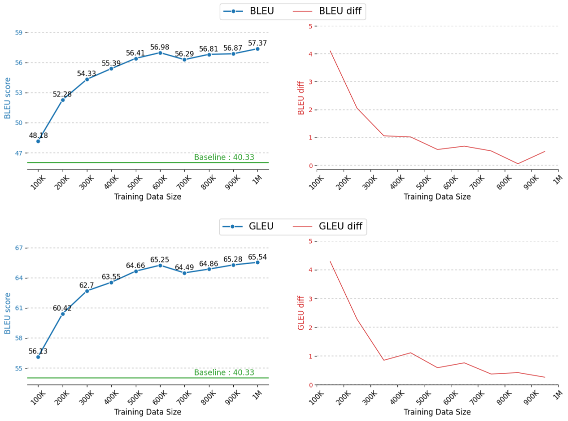

Figure 1: Performance difference depending on the amount of training data. Left figures show the performance of the Hyper-BTS-based ASR post-processor depending on the data size. Right figures (diff) show the performance difference derived by adding 100,000 training data. Baseline in each figure indicates the quality between source sentences and target sentences in Hyper-BTS test-set.

(1 million), that is, 48.18 GLEU score, and 56.13 BLEU score. These are 9.19 and 9.41 higher than the 100,000-utilized model, for the GLEU and BLEU scores, respectively. This shows that the Hyper-BTS dataset can derive sufficiently increase performance of the ASR post-processor.

One notable thing is that from 600,000 training data, the performance difference achieved by increasing the data size approximately converges to zero. This shows that there is a limit to the improvement in ASR post-processing performance that can be obtained by increasing the amount of training data. These results show similarities with back translation (Edunov et al., 2018), which is a pseudo-data generation method targeting NMT. This suggests that similar data scaling applied in NMT can be applied in the ASR post-processing field (Edunov et al., 2018) can be applied.

Effect of Parallel Corpus Filtering For the verification of the effectiveness of PCF, we did a comparative analysis of performance results with the original ASR post-processor model and the PCF applied model. Specifically, we applied PCF methodology proposed in Park et al. (2020a) to our Hyper-BTS dataset. The particular PCF method entails eliminating uncorrected aligned sentence pairs by employing the method used in Gale and Church (1993). This included pairs in which the source and target sentences are identical, which is more than 50% non-alphabetic pairs, 100 words or 1000 syllables, 30% white spaces or tabs, and a pair of sentences containing more than nine special symbols. Through these, 45,502 and 140 sentence pairs were eliminated from the training- and validation-sets, respectively.

Dataset	BLEU	GLEU
Hyper-BTS (1M)	65.54	57.37
Hyper-BTS (1M)+ Filter	**66.04 (+0.50)**	**57.45 (+0.08)**

Table 2: Parallel Corpus Filtering Effect Verification Experiment

Through the inspection of filtered data, we find that STT recognition error is the most frequent error type. By applying PCF, these errored data, as well as low quality data can be filtered out. Our experimental results considering these are shown in Table 2. These show that applying PCF can derive improvement of ASR post-processing performance. These also imply the importance of the quality of the training data and suggest the guideline for the data construction process should consider the quality of the corpus.

3.3 Qualitative Analysis

Proposal of new error types In addition to quantitative analysis, we conduct qualitative analysis. For this, we propose a new guideline for analyzing ASR post-processor trained on Hyper-BTS dataset, defining 5 primary error types as shown in Table 3.

First, we define **spacing error** as a case that there are differences in the spacing result between the recognized and reference sentence. Second, we specify **foreign word conversion error** as a case where an English word is recognized as a Korean word or vice versa. Third, we define **punctuation error** as that punctuation is not attached to the sentence or incorrectly recognized. Fourth, we define **numeric word conversion error**, where a numeric word is not recognized as a number but as a Korean word. Finally, we define **spelling and grammar error** which is the most frequent error type in ASR. Because it is a factor that strongly influences the performance of ASR systems, we subdivide it as a primary and secondary error to analyze precisely.

Primary error is defined as follows: **Deletion error** (In case that word itself, ending, or Korean postposition is not recognized.), **Addition error** (In case some syllables in a word are repeated, or unpronounced postposition or ending is added), **Substitution error** (In case that a word is replaced

Type of Error			Description	Example
Spacing error			In case the spacing result between the recognition result and correct sentence is different.	Answer: 이 불안감 뭘까 Recognized: 이불 안감 뭘까
Foreign word conversion error			In case some syllables are incorrectly converted from English to Korean or Korean to English.	Answer: SNS 이벤트 Recognized: 에스엔에스 이벤트
Punctuation error			In case some punctuation is not attached or is incorrectly used.	Answer: 밥 먹었니? Recognized: 밥 먹었니
Numeric word conversion error			In case some numbers are not converted to numbers.	Answer: 21세기 Recognized: 이십일세기
Spelling and Grammar error	Primary	Deletion	In case the whole word, Korean postposition of the word, or ending is not recognized.	Answer: 오늘 하루는 어땠어? Recognized: 하루는 어땠어?
		Addition	In case some syllables of the word are repeated, or unpronounced endings are added to the word.	Answer: 하루가 길다 Recognized: 하루하루가 길다
		Substitution	In case a word is substituted with other words which have similar pronunciation.	Answer: 순수한 사랑 Recognized: 순수한 사람
		Pronunciation Boundary	In case some words are separated or combined with the different forms between the phonetic boundaries.	Answer: 전 역시 못해요 Recognized: 저녁시 못해요
	Secondary	Spelling	In case the primary error causes a spelling error which makes the sentence nonsensical in the jamo unit.	Answer: 이제 곧 들어가야 해 Recognized: 이제 콘 들어가야 해
		Grammar	In case the primary error causes grammatical problems.	Answer: 회의 자료인 프린트 물 Recognized: 회의 자료 임프린트 물
		Meaning	In case the primary error changes the meaning of the sentence.	Answer: 21세기에 보기에는 Recognized: 21세기에 모기에는

Table 3: Error types proposed in this study for qualitative analysis of Korean ASR results. There are five main types of errors; In particular, spelling and grammar errors are subdivided into primary and secondary tagging. For these errors, both primary and secondary error tagging should be done.

with another word that has a similar pronunciation), and **Pronunciation boundary error** (In case that a word is separated into several words, or several words are combined into a single word at the boundary of pronunciation accompanied by a change in form.)

In addition, we define secondary errors as follows: **Spelling error** (In case that the primary error results in a spelling error which makes the meaning of sentence nonsensical at jamo-level), **Grammar error** (In case that the primary error causes a grammar error), **Meaning error** (In case that primary error leads to a shift in sentence meaning). If a sentence has spelling and grammar errors, both the types of primary and secondary errors defined above should be tagged.

For example, let us consider the sentence "이제 곧 들어가야 해(I have to go in soon)", recognized as "이제 콘 들어가야 해(I have to go into the corn)" by Because of misrecognition of the word '곧(soon)' as '콘(the corn),' which is a similar word but different word, a primary error is a substitution; moreover, because the entire meaning of the sentence is changed, a secondary error is meaning error.

These error types can provide the possibility of evaluating the advantages and disadvantages of the ASR model by clarifying misrecognition errors that were previously unclear in Korean speech recognition. In other words, we can summarize the weak and robust parts of various speech recognition systems by using these. Based on this criterion of errors, we also performed qualitative analysis on how well the ASR post-processor model trained with the Hyper-BTS dataset corrects which types of errors.

Main Results Table 4 shows the results of qualitative analysis of the effects of correction for each input sentence with the Hyper-BTS dataset-based ASR post-processor. This qualitatively shows that the Hyper-BTS-based post-processor can effectively correct errors that commonly occur in the Korean ASR process.

First, it was able to correct the error in which foreign words used in Korean sentences are not adequately recognized. There was a problem that the word 'David' is recognized as Korean '데이비드,' being not able to identify the foreign word. However, we exhibited that a post-processor based on the Hyper-BTS dataset could correct it.

Second, it was possible to correct the numeric word recognition error in which the numeric word is recognized as pronounced itself, not as the numeric word. As shown in the numeric conversion error in Table 4, the Hyper-BTS-based post-processor model corrected the word "열한시 (eleven o'clock)" whose pronunciation is recognized as itself to Korean, to "11시(11 o'clock)". This shows that it can provide results with improved readability by effectively correcting numeric conversion errors.

Third, it could correct the problem of attaching punctuation and spacing, which are chronic problems of the ASR model through Hyper-BTS-based post-processor. Let us consider two examples "다시한번(Onceagain)" corrected as "다시 한번(Once again)" and "괜찮을까요(is it ok)" corrected as "괜찮을까요?(is it ok?)" shown in Table 4. It indicates that the post-processor based on the Hyper-BTS dataset can supply higher quality results to the end-user by appropriately correcting erroneous spacing or punctuation cases, making users easier to under-

Input Sentence	Output Sentence	Type of Error
헌신적인 **데이비드**는 환자가 필요로 하는것은 결국 해 주겠다는것을 보여준것이 아닐까 하는 생각이 들었어요 (Devoted **데이비드**, I wondered if he was showing that he would eventually do what the patient needed)	헌신적인 **David**는 환자가 필요로 하는 것은 결국 해 주겠다는 것을 보여준 것이 아닐까 하는 생각이 들었어요. (Devoted **David**, I wondered if he was showing that he would eventually do what the patient needed.)	Foreign word conversion
그날 당신 집으로 **열한시**까지 가져다 줄게요 (I'll bring it to your house that day by **eleven o'clock**)	그날 당신 집으로 **11시**까지 가져다 줄게요. (I'll bring it to your house that day by **11 o'clock**.)	Numeric conversion
제가 교환권을 출력 하지 않았는데 괜찮을까요 (I didn't print the voucher, is it ok)	제가 교환권을 출력하지 않았는데 괜찮을까요**?** (I didn't print the voucher, is it ok**?**)	Punctuation
다시한번 나는 정말로 죄송합니다 (**Onceagain** I'm really sorry)	**다시 한번** 나는 정말로 죄송합니다. (**Once again** I'm really sorry.)	Spacing
이것은 다양한 색을 보여 주는 **사진** (**is a picture** showing the different colors)	이것은 다양한 색을 보여 주는 **사진입니다.** (**This is a picture** showing the different colors.)	Spelling and Grammar errors (Deletion-Meaning)
그것은 범죄 사건이 일어난 지역 소속의 공정한 배심원단에 의하여 **진행된** (This **was decid** by an impartial jury from the area where the crime took place)	그것은 범죄 사건이 일어난 지역 소속의 공정한 배심원단에 의하여 **진행됩니다**. (This **is decided** by an impartial jury from the area where the crime took place)	Spelling and Grammar errors (Deletion-Grammar)
우리집엔 좋은 경치를 가지고 있어요 (**In ourhouse** has a nice view)	**우리 집은** 좋은 경치를 가지고 있어요. (**Our house** has a nice view.)	Spelling and Grammar errors (Substitution-Grammar)
이진훈이 제일 우선이라는 걸 명심하세요 (Keep in mind that **LeeJinHoon** has priority)	**이 주문**이 제일 우선이라는 걸 명심하세요. (Keep in mind that **this order** has priority.)	Spelling and Grammar errors (Pronunciation Boundary-Meaning)

Table 4: Examples of Hyper-BTS dataset-based ASR post-processor outputs for qualitative analysis. Note that we indicate text containing the corresponding errors generated by BTS in red; also, we indicate the original correct result in blue text.

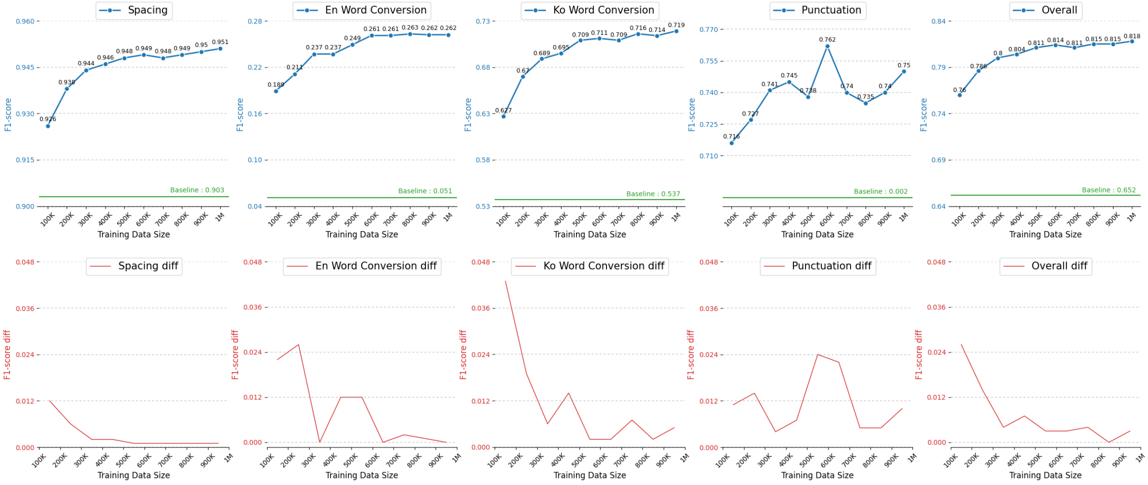

Figure 2: Performance difference, depending on the amount of training data. F1-scores are reported for each feature, including model performance on automatic spacing, word conversion, punctuation, and overall. KO and EN indicate Korean English respectively. Upper figures show the performance of the Hyper-BTS-based ASR post-processor for the above three factors, depending on the data size. Lower figures show the performance difference (diff) for the above three factors, derived by utilizing additional 100,000 training data. Baseline indicates the f1-score of each factor between source sentences and target sentences in Hyper-BTS test-set.

stand the intent of the sentence.

Fourth, the Hyper-BTS dataset-based post-processor model was able to correct word substitution caused by speech recognition errors with similar pronunciations or word separation and integration problems at pronunciation boundaries between the words in consideration of surrounding contexts. In the example sentence of Table 4, Hyper-BTS-based post-processor corrected "우리집엔(In ourhouse)" which is a substitution-grammar error as "우리 집은(Our house is)." It can be said that the post-processor corrected the adverb Korean postposition "-엔(In)" to nominative postposition "-은(is)" which can make the word nominative, considering the grammatical information. In the following example sentence, the subject recognized as "이진훈(LeeJinHoon)", which means a person's name, was corrected to "이 주문(This order)" regarding the context of the ordinal information of "우선(priority)."

Fifth, it can be confirmed that the Hyper-BTS dataset-based post-processor plays a significant role in correcting sentences that are not attached adequately with terminating endings because of speech recognition errors, filling the incompleteness of the sentence structure. In Korean, an error in which the ending is not appropriately attached is a problem that must be resolved because it dramatically changes the meaning of a sentence beyond a spelling error.

In particular, because of the head-final linguistic characteristics of Korean, where the predicate is placed at the end of the sentence, if sentence termination is not done correctly, the sentence's overall semantic and syntactic structure can be significantly changed. As shown in the example sentence with the deletion error in Table 4, even the cases that the syntactic structure of the sentence was changed because of the disappearance of "입니다(is)" at the end of the sentence, it was possible to correct it as a complete sentence by restoring some part of omitted. Also, the word "진행된(decid)," which is caused an error by recognition error of the terminating ending, could be corrected as "진행됩니다(decided)" with the appropriate terminating.

Additionally, we analyzed correction effects of post-processor according to the amount of trained data in Appendix B.

3.4 Additional Analysis

In this experiment, we analyzed the practical effectiveness of Hyper-BTS-based ASR post-processor with the following three aspects: Spacing, Foreign word conversion, Punctuation. These are mainly related to the readability and satisfaction of the end users of the ASR services.

As in the previous experiment, we established 10 anchor points to the whole training data, and verified the performance difference induced by increasing the data size. We inspected the corrected sentence by checking whether each factor is in the correct position. For the performance evaluation of each post-processor, corresponding multi-class accuracy is estimated based on the f1-score. Experimental results are shown in Figure 2.

Automatic Spacing We first evaluated the practical effectiveness that can be obtained by applying Hyper-BTS-based ASR post-processor. As can be seen in our figure, a generally larger amount of data derived higher performance, and performance difference goes to converge as adding more training data. Especially, the performance of the post-processor trained by 1M data shows a 0.951 f1-score, which indicates that spacing errors can almost be thoroughly corrected by our Hyper-BTS-based ASR post-processor.

Foreign Word Conversion For the evaluation of the foreign word conversion, we counted the number of correct positions of Korean and English words in a target sentence and estimated f1-score. Through our experiments, it can be seen that ASR post-processor attained 0.182 and 0.211 f1-score higher performance than the baseline, for the Korean word and English word conversion, respectively. Considering English word conversion, baseline showed a 0.0051 f1-score, which shows the weak point of the ASR system. However, this can be effectively amended by ASR post-processor, up to 0.262 f1-score.

Punctuation Attachment Considering punctuation attachment, we used f1-scores that check the correct position of the symbols in a target sentence. As shown in the fourth plot of the Figure 2, we can find that the baseline shows only a 0.002 f1-score. This indicates that the punctuation attachment, symbol attachment, and sentence separation can be seen as some of the most challenging issues of the ASR system. However, we can find that the f1-score about the punctuation attachment can be raised up to 0.762 by applying ASR post-processor, and even with 100K training data, we can obtain a 0.715 f1-score. This result shows that Hyper-BTS-based post-processor can effectively deal with the inherent limitations of the ASR system.

Overall f1-score Finally, we verified the effectiveness of the Hyper-BTS dataset for the overall performance of the above factors. As can be seen in the results, compared with the baseline which shows a 0.652 f1-score, post-processing can improve its quality up to 0.818. This shows that the Hyper-BTS-based ASR post-processor can effectively catch and correct the internal errors that ASR system cannot deal with.

4 Conclusion

In this study, we conducted a thorough analysis of results from rigorous experiments after developing the Hyper-BTS dataset and training an Automatic Speech Recognition (ASR) post-processor. Both quantitative and qualitative outcomes validate the effectiveness of the Hyper-BTS dataset in enhancing the performance of the ASR post-processor. Recognizing the broader implications of our research, we are committed to facilitating unrestricted access to this dataset for both industry professionals and academic researchers. Additionally, we pioneered a robust quality control mechanism by formulating novel guidelines anchored in the categorization of ASR post-processor error types, thereby aiming to elevate the qualitative dimensions of ASR post-processing.

Acknowledgments

This work was supported by ICT Creative Consilience Program through the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT)(IITP-2024-2020-0-01819). This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2021R1A6A1A03045425). This work was supported by Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2022-0-00369, (Part 4) Development of AI Technology to support Expert Decision-making that can Explain the Reasons/Grounds for Judgment Results based on Expert Knowledge)

Limitations

While our research primarily focuses on the Korean language, the depth of this investigation offers significant insights even within this narrow scope. Nevertheless, we understand the importance of expanding to other languages in subsequent studies.

A limitation of our current study is the use of the Vanilla Transformer for our experiments. We chose this model to evaluate the Hyper-BTS dataset due to its broad use and manageable computational requirements, especially when compared to cutting-edge models. By using the Vanilla Transformer, we aimed to present findings that are both practical in terms of computational cost and relevant to a wide range of researchers.

Most importantly, we would like to clarify the key contributions of our work. In this study, we built a large scale ASR post-processing datasets (Hyper-BTS) that has shown to significantly improve the performance of ASR post-processors as shown in Figure 1 & 2. On top of releasing the dataset, we believe that our use of BTS technology in curation of the dataset is also a significant contribution as it shows how large-scale parallel corpora can be created effortlessly, without any form of human annotation. Furthermore, our dataset creation process only requires raw Korean textual data to train ASR post-processor which is arguably more abundant than other forms of data such as GEC (Grammar Error Correction) Korean text dataset.

In addition to the dataset, we also propose a new guideline to analyzing Korean ASR results through definition of different error types as shown in Table 3. With the new analysis guideline, which was previously unavailable for Korean ASR, and along with the newly proposed Hyper-BTS dataset, we hope to benefit other researchers in this field of research.

Ethics Statement

Hyper-BTS is built using datasets publicly available online on platforms such as Korean AI-HUB. These datasets are open-source and free from copyright issues. As such, after thorough examination of our dataset curation and experimentation procedure, we are confident that there is no ethical issue in our work. Also, We reviewed all ethical issues in our experiments and made fair comparisons.

References

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Joon Son Chung. 2019. Naver at activitynet challenge 2019–task b active speaker detection (ava). *arXiv preprint arXiv:1906.10555*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

William A Gale and Kenneth Church. 1993. A program for aligning sentences in bilingual corpora. *Computational linguistics*, 19(1):75–102.

Mark Gales and Steve Young. 2008. The application of hidden markov models in speech recognition.

Manuel Giollo, Deniz Gunceler, Yulan Liu, and Daniel Willett. 2020. Bootstrap an end-to-end asr system by multilingual training, transfer learning, text-to-text mapping and synthetic audio. *arXiv preprint arXiv:2011.12696*.

Nils Hjortnæs, Niko Partanen, Michael Rießler, and Francis M Tyers. 2021. The relevance of the source language in transfer learning for asr. In *Proceedings of the Workshop on Computational Methods for Endangered Languages*, volume 1, pages 63–69.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Guillaume Klein, Dakun Zhang, Clément Chouteau, Josep M Crego, and Jean Senellart. 2020. Efficient and high-quality neural machine translation with opennmt. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 211–217.

Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzmán. 2020. Findings of the wmt 2020 shared task on parallel corpus filtering and alignment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 726–742.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Junwei Liao, Sefik Emre Eskimez, Liyang Lu, Yu Shi, Ming Gong, Linjun Shou, Hong Qu, and Michael Zeng. 2020. Improving readability for automatic speech recognition transcription. *arXiv preprint arXiv:2004.04438*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Anirudh Mani, Shruti Palaskar, Nimshi Venkat Meripo, Sandeep Konam, and Florian Metze. 2020. Asr error correction and domain adaptation using machine translation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6344–6348. IEEE.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 588–593.

OpenAI. 2023. Gpt-4 technical report.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Chanjun Park, Sugyeong Eo, Hyeonseok Moon, and Heui-Seok Lim. 2021a. Should we find another model?: Improving neural machine translation performance with one-piece tokenization method without model modification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 97–104.

Chanjun Park, Yeonsu Lee, Chanhee Lee, and Heuiseok Lim. 2020a. Quality, not quantity? : Effect of parallel corpus quantity and quality on neural machine translation. In *The 32st Annual Conference on Human & Cognitive Language Technology*, pages 363–368.

Chanjun Park, Jaehyung Seo, Seolhwa Lee, Chanhee Lee, Hyeonseok Moon, Sugyeong Eo, and Heuiseok Lim. 2021b. BTS: Back TranScription for speech-to-text post-processor using text-to-speech-to-text. In *Proceedings of the 8th Workshop on Asian Translation (WAT2021)*, pages 106–116, Online. Association for Computational Linguistics.

Chanjun Park, Yeongwook Yang, Kinam Park, and Heuiseok Lim. 2020b. Decoding strategies for improving low-resource machine translation. *Electronics*, 9(10):1562.

Matthew Nicholas Stuttle. 2003. *A Gaussian mixture model spectral representation for speech recognition*. Ph.D. thesis, University of Cambridge.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Kimberly Voll, Stella Atkins, and Bruce Forster. 2008. Improving the utility of speech recognition through error detection. *Journal of digital imaging*, 21(4):371.

Zi-Qiang Zhang, Yan Song, Ming-Hui Wu, Xin Fang, and Li-Rong Dai. 2021. Xlst: Cross-lingual self-training to learn multilingual representation for low resource speech recognition. *arXiv preprint arXiv:2103.08207*.

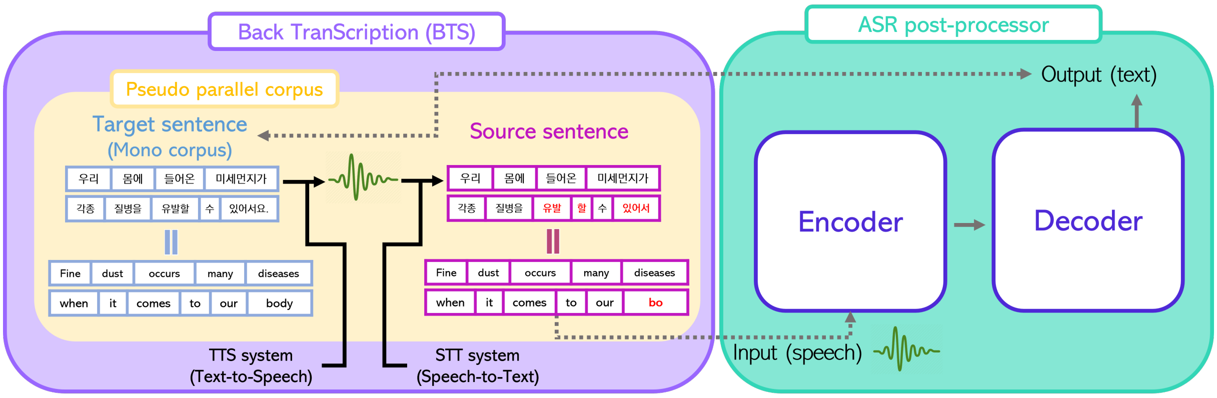

Figure 3: Architecture of the ASR post-processor and BTS for building Hyper-BTS dataset. The red-colored words in the source sentence indicate ungrammatical words. The following example means "Fine dustoccurs many diseases when it comes to our bo" from the source sentence and means "fine dust occurs many diseases when it comes to our body" from the target sentence.

A What is BTS?

BTS is a self-supervised method that automatically constructs the training dataset for the S2S-based ASR post-processor (Park et al., 2021b). BTS can easily obtain pre-built mono corpus using crawling; the collected corpus is automatically transformed into a parallel pair without human labor by converting the text files into voice files through the TTS system and subsequently reproducing the generated voice files to text files through the STT system. It consists of target sentences acquired from the mono corpus and source sentences that go through a round trip process that converts target sentences back to text via the TTS and STT. Finally, the ASR post-processor model can be constructed using the machine-generated pseudo-parallel corpus as a training dataset.

Figure 3 demonstrates the structure of the BTS and the learning process of the speech recognition post-processor model based on the S2S using the derived dataset. We reproduced this architecture following BTS procedure.

As illustrated in Figure 3, the overall architecture is given in the following: (BTS module) – TTS system converts the target sentence (gold sentence) into speech. Subsequently, the speech is transferred to STT system, which makes the source text (ungrammatical sentence). (ASR post-processor module) – this module conducts S2S training, where uses a speech from the source sentence for the input and the target sentence as a ground truth.

BTS can build the training data infinitely. Despite the disadvantages of building a parallel corpus, such as time, money, and accessibility, BTS has the advantage of building an interminable mono corpus through the website. From this policy, it is possible to build unlimited training data and enable boosting the building of our Hyper-BTS dataset. Furthermore, it can solve the limitations (*i.e.,*spacing, foreign conversion, punctuation, grammar correction) of the existing speech recognition system as a universal model since the mono corpus used as the target sentence is primarily free of this problem.

Furthermore, it is a method that does not require the role of a phonetic transcriptor and has tremendous advantages in terms of time and cost. In addition, there is an advantage of being free from problems regarding the quality difference between phonetic transcriptors.

For Park et al. (2021b), the language pair for the BTS experimentation was set to Korean. Finally, a pseudo parallel corpus of 229,987 sentence pairs for the S2S-based ASR post-processor was constructed by BTS.

B Qualitative analysis according to the amount of training data (100k VS 1M)

As shown in Table 5, we classified the Hyper-BTS dataset into cases of 100K training data and 1M training data, respectively, and analyzed the post-editing results for the same test input. Through this analysis, we confirm the effect of ASR post-processing according to the size of our proposed Hyper-BTS dataset.

Input Sentence	**Output Sentence**	
	Hyper-BTS$_{100K}$	**Hyper-BTS**$_{1M}$
미안한데 **시디**만 따로 보내 주실 수 있나요 (Excuse me, can you send the **시디** separately)	미안한데 **시디**만 따로 보내주실 수 있나요? (Excuse me, can you send the **시디** separately?)	미안한데 **CD**만 따로 보내주실 수 있나요? (Excuse me, can you send the **CD** separately?)
그리고 **두명**의 학생을 위해서 2개 기숙사 방을 예약하고 싶습니다 (And I would like to reserve 2 dormitory rooms for **two** students)	그리고 **두 명**의 학생을 위해서 2개의 기숙사 방을 예약하고 싶습니다. (And I would like to reserve 2 dormitory rooms for **two** students.)	그리고 **2명**의 학생을 위해서 2개의 기숙사 방을 예약하고 싶습니다. (And I would like to reserve 2 dormitory rooms for **2** students.)
왜 이렇게 일찍 일어났어요 (Why did you wake up so early)	왜 이렇게 일찍 일어났어요**.** (Why did you wake up so early**.**)	왜 이렇게 일찍 일어났어요**?** (Why did you wake up so early**?**)
이 근처에서 볼 수 **있는데가** 있어요 (There are **placesto** watch around here)	이 근처에서 볼 수 **있는데가** 있어요. (There are **placesto** watch around here.)	이 근처에서 볼 수 **있는 데가** 있어요. (There are **places to** watch around here.)
성적서는 7월 25일까지 발급 **되면** (**If** the certificates **are** issued by July 25th)	성적서는 7월 25일까지 발급**되면?** (**If** the certificates **are** issued by July 25th?)	성적서는 7월 25일까지 발급**됩니다.** (Certificates **will be** issued by July 25th.)

Table 5: Examples of the correction result according to the amount of training data. Note that we indicate the text where includes errors in red; also, we indicate the miscorrected text by Hyper-BTS in the same color. In addition, we indicate the text corrected properly by Hyper-BTS in blue.

First, Hyper-BTS$_{1M}$ shows better correction of foreign language conversion errors. Hyper-BTS$_{100K}$ does not correct the foreign word “CD (CD)’, whereas Hyper-BTS$_{1M}$ corrects it properly. Second, in the case of “두명(two)", which has not been converted to a word containing numbers, Hyper-BTS$_{100K}$ recognizes it as a space error and corrects it with “두 명(two)". Hyper-BTS$_{1M}$ shows more robust results in numeric conversion correction by successfully correcting “2명". Third, Hyper-BTS$_{100K}$ recognizes a punctuation error for the sentence “왜 이렇게 일찍 일어났어요(Why did you wake up so early)", but adds punctuation in the declarative form instead of the interrogative one. On the other hand, Hyper-BTS$_{1M}$ successfully correct punctuation and show effectiveness in the punctuation area. Fourth, Hyper-BTS$_{1M}$ is more effective as a result of correction for spacing errors, which is an important factor in readability. Although Hyper-BTS$_{100K}$ fail to correct “볼 수 있는데가(placesto)", Hyper-BTS$_{1M}$ successfully post-edit to “볼 수 있는 데가(places to)". Finally, in the deletion error, which is a representative and important error of spelling and grammar due to misrecognition of the main predicate, Hyper-BTS$_{100K}$ corrects the ending that is not properly terminated into an interrogative sentence as it is. Whereas, Hyper-BTS$_{1M}$ shows the result of successful correction considering the context.