mathematics

Article

The ASR Post-Processor Performance Challenges of BackTranScription (BTS): Data-Centric and Model-Centric Approaches

Chanjun Park [1,2], Jaehyung Seo [1], Seolhwa Lee [3], Chanhee Lee [4] and Heuiseok Lim [1,*]

[1] Department of Computer Science and Engineering, Korea University, Seoul 02841, Korea
[2] Upstage, Gyeonggi-do 16942, Korea
[3] Department of Computer Science, University of Copenhagen, DK-2100 Copenhagen, Denmark
[4] Naver Corporation, Gyeonggi-do 13561, Korea
[*] Correspondence: limhseok@korea.ac.kr

Abstract: Training an automatic speech recognition (ASR) post-processor based on sequence-to-sequence (S2S) requires a parallel pair (e.g., speech recognition result and human post-edited sentence) to construct the dataset, which demands a great amount of human labor. BackTransScription (BTS) proposes a data-building method to mitigate the limitations of the existing S2S based ASR post-processors, which can automatically generate vast amounts of training datasets, reducing time and cost in data construction. Despite the emergence of this novel approach, the BTS-based ASR post-processor still has research challenges and is mostly untested in diverse approaches. In this study, we highlight these challenges through detailed experiments by analyzing the *data-centric* approach (i.e., controlling the amount of data without model alteration) and the *model-centric* approach (i.e., model modification). In other words, we attempt to point out problems with the current trend of research pursuing a model-centric approach and alert against ignoring the importance of the data. Our experiment results show that the data-centric approach outperformed the model-centric approach by +11.69, +17.64, and +19.02 in the F1-score, BLEU, and GLEU tests.

Keywords: backtranscription; machine translation; data-centric; model-centric; automatic speech recognition; post-processor

MSC: 68T50

check for updates

Citation: Park, C.; Seo, J.; Lee, S.; Lee, C.; Lim, H. The ASR Post-Processor Performance Challenges of BackTranScription (BTS): Data-Centric and Model-Centric Approaches. *Mathematics* **2022**, *10*, 3618. https://doi.org/10.3390/math10193618

Academic Editor: Qingshan Jiang

Received: 17 August 2022
Accepted: 29 September 2022
Published: 2 October 2022

1. Introduction

Automatic speech recognition (ASR) is an automation system that converts human voices into text. Based on traditional studies on speech recognition architectures, such as Gaussian mixture models [1] and hidden Markov models [2], recent ASR research studies have conducted transfer learning on pre-trained models [3–5]. In other words, studies on ASR have been conducted to improve speech recognition performance through model-centric approaches with model modifications.

However, despite the success of these model-centric ASR studies, limitations regarding real-world applications still exist [6]. This approach requires an appropriate service circumstance with sufficient computing power (e.g., GPUs) to process large-scale resources. The model-centric approach demands a tremendous amount of parameters and datasets to train the modified models, making it challenging for companies with insufficient computing resources or GPU environments to configure their services using these state-of-the-art models.

Contrary to the model-centric approach, research on the data-centric approach indicates that the quality and amount of data can increase without model modifications, and improvement of ASR models can be achieved through pre-processing and post-processing [7–11]. The data-centric approach can be applied to any lightweight model that is untroubled to proceed with the ASR service and can be combined into a model that is fully serviceable with only CPUs [12], such as the vanilla transformer [13], thereby mitigating the aforementioned limitations.

Recently, Park et al. [14] proposed an ASR post-processor method, BackTranScription (BTS), based on the data-centric approach. BTS is a data-building methodology that can alleviate the limitations of existing sequence-to-sequence (S2S)-based ASR post processors and can combine text-to-speech (TTS) with speech-to-text (STT) to generate a pseudo-parallel corpus.

To improve the performance of BTS-based ASR post-processors, we evaluate the model-centric approach and data-centric approach settings on a key basis and conduct comparative experiments. In this study, we utilize a copy mechanism in the model-centric approach in accordance with the source language and the target language character set (i.e., language) being the same and interpret the decreasing performance. We implement experiments derived from the same model used in Park et al. [14] to evaluate the effects of the data-centric approach and double both the quantitative and qualitative aspects of the dataset using parallel corpus filtering (PCF) [15].

2. What is BTS?

BTS is a self-supervised method that automatically constructs the training dataset for the S2S-based ASR post-processor [14]. We use a crawler to obtain a pre-designed monolingual corpus; the gathered corpus is mechanically turned into a parallel corpus without human effort by transforming the text into voice files through the TTS system and reproducing the generated voice-to-text files in the STT system. BTS contains target sentences acquired from the gathered corpus and source sentences through a round-trip process that converts target sentences back to text via the TTS and STT. After that, the model-generated pseudo-parallel corpus will be a training dataset for the ASR post-processor model.

The overall architecture is as shown in Figure 1. (1) (BTS module)—the TTS system transforms the target sentence (gold sentence) into speech. (2) The speech is fed to the STT system to make the source text (grammatically damaged sentence). (3) (ASR post-processor module)—this module operates S2S training, where it employs the speech from the source sentence for the input and target sentence as a ground truth.

The traditional dataset construction approach has the drawbacks of producing a parallel corpus, such as accessibility, time, and money. However, BTS can produce the training data infinitely. Additionally, it retains the benefit of making an interminable monolingual corpus through the website. Sequentially, we can solve the restrictions (i.e., spacing, foreign conversion, punctuation, grammar correction) of the current speech recognition system as a universal model.

Additionally, it is a method that reduces the role of a phonetic human transcriptor and has immense benefits in terms of time and cost. Moreover, there is little difference between phonetic transcriptions presented by humans and sentences post-processed by the model.

We set the language pair for the experimentation to a low resource language considering Park et al. [14]. Then, 129,987 sentences provided as Korean scripts in the business and technology TED were selected using web crawling for data construction. In addition, we extracted 105,000 sentences from the AI-HUB Korean–English translation (parallel) corpus. Collected raw sentences were subsequently converted into mp3-format speech data operating the Google TTS API. This process was accomplished to lower the entry barrier by enabling BackTranscription to be used by companies that do not have an internal TTS system, which would make the system suitable for commercial use. Therefore, this research also used Google API as the TTS for BTS.

After that, it was converted back to text data using the NAVER CLOVA speech recognition (CSR) API based on the voice data built through TTS. Lastly, a pseudo-parallel corpus of 229,987 sentence pairs for the S2S-based ASR post-processor was constructed. This study also used the CLOVA speech recognition API as the STT for BTS.

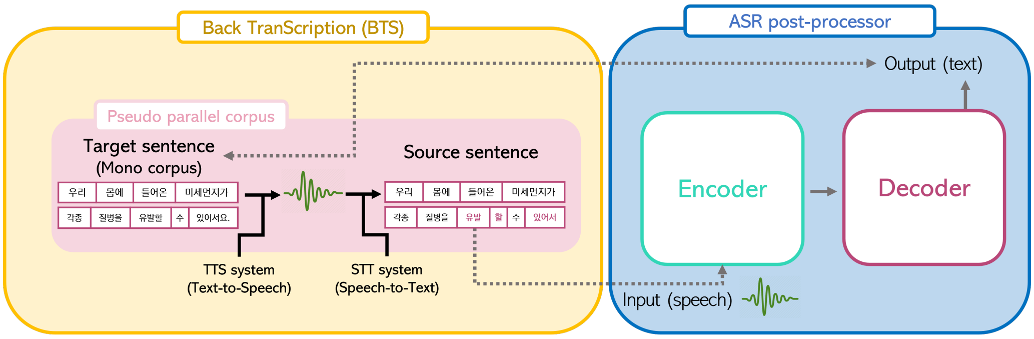

Figure 1. Architecture of the ASR post-processor and BTS. The red-colored words in the source sentence indicate ungrammatical words. The following example means "Fine dustoccurs many diseases when it comes to our bo" from the source sentence and means "fine dust occurs many diseases when it comes to our body" from the target sentence.

3. Experimental Setup

3.1. Model-Centric Approach

To enhance the performance of the ASR post-processors based on BTS, we conducted experiments on the model-centric approach, which sought to achieve improvements through model modifications. We trained our model on the vanilla transformer [13], which further applied the copy mechanism to validate whether or not the model modification improved the performance of the model.

Copy Mechanism for BTS

The copy mechanism [16] dynamically copies the words from the source text while decoding it to solve the out-of-vocabulary (OOV) problem where words are needed to generate sentences. However, the copy mechanism is still unable to extract the proper embeddings of the OOV from the input context. Thus, we approach copy attention, which integrates both the existing attention mechanism and the copy mechanism to resolve them and calculate the probabilities to capture whether to copy or not. Specifically, copy attention can calculate as formula: $p(w) = p(z = 1)p_{copy}(w) + p(z = 0)p_{softmax}(w)$, where $p_{softmax}$ is the standard softmax over the target dictionary, $p(z)$ is the probability of copying a word from the source, and p_{copy} is the probability of copying a particular word taken from the attention distribution directly.

This is achieved in two states: (encoder state)—the probability of outputting the vocabulary with the highest copy attention score among the input sequences and (decoder state)—the probability of the vocabulary in the output vocabulary dictionary occurring when predicting the output vocabulary for each time-step during the decoding process.

In this study, we applied the copy mechanism according to the characteristics of the training data in the BTS, where the source and target sentences had the same character set. Thus, we handled the OOV using the copy mechanism in a principled manner. Subsequently, we employed three different attention functions—*dot* [17], *bilinear* [17], and *multi-layer perceptron (MLP)* [18] operations to compute the attention score and explore performance comparisons.

The hyperparameters were set to the same values used in the vanilla transformer model employed in the prior models [13]. Furthermore, the vocabulary size was 32 k, and we used SentencePiece [19] for the subword tokenization.

3.2. Data-Centric Approach

We implemented the data-centric approach to evaluate the change in performance by controlling the amount and quality of the dataset without model modifications [20,21].

In a prior study employing BTS, Park et al. [14] set the language pairs to Korean as a low resource language and collected the monolingual corpus from two main sources for data construction (AIHUB [22], TED). The study conducted training based on a pseudo-parallel corpus dataset consisting of 219,318 sentences and achieved reasonable improvement.

However, the size of the dataset still lacked compared to the parallel corpus used by recent neural machine translation studies.

Therefore, we further validated the performance improvement by building an additional 500,000 pseudo-parallel pairs, which was more than twice the amount of the existing dataset (*No-Filter*). Further, we conducted experiments to filter out the unnecessary noise by employing PCF (*Filter*).

PCF helps build a fine quality parallel corpus and screens for sentences that are in an acceptable condition. For the pseudo-parallel corpus built over the BTS, unrecognized sources or target sentences, and excessively long or short outliers caused by unintended errors in the STT and TTS system, remain as drawbacks. Therefore, we filtered out 14,497 sentences by applying the PCF proposed in Park et al. [23] to ensure a high-quality dataset. Park et al. [23] proposed the method, which eliminates the uncorrected aligned sentence pairs by employing the method used in Gale and Church [24], including 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.

After this proposed process was completed, we obtained the following filtered values: more than 100 words or 1000 syllables in one pair and 27 misaligned sentence pairs. We also obtained the results for more than 30% white spaces or tabs, which were 10,301 pairs, and more than 50% non-alphabetic pairs, which were 4165. Additionally, three pairs of empty sources or targets were filtered out. This was because of a recognition error during the STT process.

Based on these datasets, which were refined through filtering, we control the amount and quality of the dataset without model modification to verify whether the data-centric approach improves the performance of the model.

To make the results comparable to previous work, we introduced the additional examples into the training split without modifying the validation and test splits.

4. Experimental Results

4.1. Main Results: Data-Centric or Model-Centric?

We evaluated the model through the general language evaluation understanding (GLEU) [25] and bilingual evaluation understudy score (BLEU) [26] metrics. The results of the conducted experiments are shown in Table 1. We used the results of Park et al. [14] as the baseline for the comparisons.

As shown in Table 1, the baseline for the BLEU and GLEU scores are 56.56 and 46.92, respectively. For the model-centric approach, the average performance degradation of the BLEU and GLEU scores are 7.97 and 6.96, respectively. In other words, *in many studies, the performance of the attention mechanisms usually improved when the copy mechanism was applied; however, in this study, we found that the performance of all three attention mechanisms decreased.*

Despite the character set being the same on the source and the target side, this result could be interpreted as there being cases where completely different languages or character sets are mixed, such as a non-uniformity of numbers (e.g., two, 2) and foreign word conversion (e.g., Oprah Winfrey/오프라 윈프리)

Additionally, another interpretation is grammatically incorrect sentences or sentences that were not complete caused an unknown problem. Furthermore, the result could have also been caused by the inconsistency in tokenization with regards to the spacing error on the source text, which could adversely affect the copy mechanism.

Nonetheless, for the data-centric approach, we found that the performance improved significantly. We simply conducted the large-scale augmentation method to double the amount of data, resulting in improved BLEU and GLEU sores, which were 8.73 and 11.58, respectively. Through this result, we could determine the necessary amount of data. The data-centric approach outperformed the baseline vanilla transformer model with significant improvement across all conditions, which was the opposite of what was observed for the model-centric approach.

In addition, it was found that the BLEU and GLEU scores improved by 9.87 and 12.26, respectively, when we applied the filtering, compared to the study conducted by Park et al. [14]. These results show that the BLEU score and GLEU score (1.14 and 0.68, respectively) improved as compared to the no-filter model. Thus, we can infer that the quantity of data is important and that the quality factor must also be considered.

In conclusion, the data-centric approach performed overwhelmingly better than the model-centric approach.

Table 1. Verification of data and model-centric approach.

Centric	Model	BLEU	GLEU
-	Park et al. [14]	56.56	46.94
Model	Copy-*MLP*	48.79 (−7.77)	39.91 (−7.03)
Model	Copy-*dot*	48.71 (−7.85)	40.18 (−6.76)
Model	Copy-*bilinear*	48.25 (−8.31)	39.83 (−7.11)
Data	No-Filter	65.29 (+8.73)	58.52 (+11.58)
Data	Filter	**66.43 (+9.87)**	**59.20 (+12.26)**

4.2. Insights from the Negative Results of the Model-Centric Approach

We analyzed three significant aspects for the negative results and inferred from them regarding the model-centric approach. First of all, we analyzed the model inference speed to gauge the efficiency of the model. The vanilla transformer took 0.009 s per sentence to perform grammar corrections and processed 1622.17 tokens per second. While applying the copy mechanism with the vanilla transformer, it took 0.0114 s to perform grammar corrections per sentence and processed 1309.27 tokens per second; however, it increased the number of parameters. In other words, these results reveal that the size of the model is bumped when we implement model modifications, which causes the inference speed to decrease.

Second, we investigated the performance of the model to analyze its effectiveness. As can be seen from the results in Table 1, the performance of the model did not improve but rather worsened. Although the result is only in regards to the BTS, this suggests that model modification is not the optimal choice for any of the cases.

Finally, we examined the implementation settings, which may constitute another major drawback. The model can be a hindrance for small companies that lack the hardware to provide the service owing to the vast parameters used and the size of the model. In other words, having many parameters and building large-capacity models is not optimal for providing an efficient service.

4.3. Additional Analysis

Furthermore, we conducted additional verification on three factors: spacing [27], foreign word conversion, and punctuation [28] to gauge the readability and satisfaction of the ASR service for end-users. We used the F1-score for the performance metric for all three factors. F1-score measures the average overlap between the post-processed sentence and ground truth target sentence. We treat the post-processed and the ground truth sentences as bags of tokens, similar to the evaluation method of SQuAD [29].

The experimental result is shown in Table 2. These results show that the data-centric approach improves the performance better than the model employed in the study of Park et al. [14]; whereas, the performance of the model-centric method deteriorates. This shows that conducting filtering on the data-centric approach results in a higher performing model than the non-filtering model for all factors except word conversion (EN).

Table 2. Comparison between model-centric and data-centric approaches: F1-scores are reported for each feature, including model performance on automatic spacing, word conversion, punctuation, and overall. KO: Korean, and EN: English.

Model	Type	Spacing	Word Conversion (KO)	Word Conversion (EN)	Punctuation	Overall
Park et al. [14]	Baseline	91.86	54.41	23.41	61.02	70.73
Copy-*MLP*	Model-Centric	91.09 (−0.77)	47.23 (−7.18)	15.19 (−8.22)	56.78 (−4.24)	66.40 (−4.33)
Copy-*dot*	Model-Centric	91.24 (−0.62)	47.42 (−6.99)	15.14 (−8.27)	58.90 (−2.12)	66.40 (−4.33)
Copy-*bilinear*	Model-Centric	91.24 (−0.62)	47.42 (−6.99)	15.13 (−8.28)	58.90 (−2.12)	66.72 (−4.01)
No-Filter	Data-Centric	94.58 (+2.72)	65.27 (+10.86)	**44.81 (+21.40)**	74.96 (+13.94)	78.40 (+7.67)
Filter	Data-Centric	**94.59 (+2.73)**	**65.41 (+11.00)**	40.36 (+16.95)	**75.25 (+14.23)**	**78.41 (+7.68)**

Consequently, we derived an overall F1-score for the integrated performance of all factors. The filtering of the data-centric approach model depicted the best score, which was 78.41.

4.4. Qualitative Analysis

We conducted a qualitative analysis of the error types mentioned in Table 2. We set sentences including errors occurring for user input as source text due to the limitation of speech recognition in the ASR system. Table 3 shows the post-processing results of the source text with errors. Copy-bilinear and Filter were used as the comparative model, which showed the highest performance on average in model-centric and data-centric, respectively.

Table 3. Examples of model-centric and data-centric outputs for qualitative analysis.

Input Sentence (Source Text with Errors)	Copy-Bilinear (Model-Centric)	Filter (Data-Centric)	Reference Sentence
강남역에는 다양한 **레스토랑 카페 쇼핑 샵들이** 많아 항상 수많은 젊은이들로 **붐 빕니다** (Gangnam Station is always **crowd ed** with many young people because there are many **restaurants cafes shopping shops**) Spacing & Punctuation Error	강남역에는, 다양한 **레스토랑 카페 쇼핑 샵들이** 많아 항상 수많은 젊은이들로 **붐_빕니다**. (Gangnam Station is always **crowd_ed** with many young people because there are many **restaurants cafes shopping shops**.)	강남역에는 다양한 **레스토랑, 카페, 쇼핑 샵들이** 많아 항상 수많은 젊은이들로 **붐빕니다.** (Gangnam Station is always **crowded** with many young people because there are many **restaurants, cafes, and shopping shops.**)	강남역에는 다양한 레스토랑, 카페, 쇼핑 샵들이 많아 항상 수많은 젊은이들로 붐빕니다. (Gangnam Station is always crowded with many young people because there are many restaurants, cafes, and shopping shops.)
얼마전까진 **의사**와의 회의 결과 본 계약 예약 시점은 약간 지연 된다고 합니다 (Not too long ago according to a meeting with **the doctor** the reservation time for this contract will be delayed a little.) Word Conversion (KO) & Punctuation Error	얼마 전까진 **의사**와의 회의 결과 본 계약 시점은 약간 **지연고** 합니다. (Not long ago as a result of the meeting with **the doctor** the timing of this contract has been **dely** a little.)	얼마 전 가진 **A사**와의 회의 결과**,** 본 계약 예약 시점은 약간 지연된다고 합니다. (As a result of a recent meeting with **A company,** the reservation time for this contract is slightly delayed.)	얼마 전 가진 A사와의 회의 결과, 본계약의 계약 시점은 약간 지연된다고 합니다. (As a result of a recent meeting with A company, the contract timing of this contract is slightly delayed.)
먹방은 먹는 방송이라는 한국 말의 줄임말로 한국방송 **개** 혜성처럼 등장한 새로운 **trend hair** (Mukbang is short for eating show in Korean and it's a new **trend hair** that has emerged like a Korean broadcasting **dog** comet.) Word Conversion (EN) & Weird Word Error	먹방은 먹는 방송이라는 한국 말의 줄임말로 한국방송을 혜성처럼 등장한 새로운 **Th-에도**요. (Mukbang is short for eating show and it's a new **Th-that** appeared like a comet.)	먹방은 먹는 방송이라는 한국말의 줄임말로 한국 방송계에 혜성처럼 등장한 새로운 **트렌드예요.** (Mukbang is a new **trend** that has emerged like a comet in the Korean broadcasting industry, short for eating broadcasting).	먹방은 먹는 방송이라는 한국말의 줄임말로 한국 방송계에 혜성처럼 등장한 새로운 트렌드예요. (Mukbang is a new trend that has emerged like a comet in the Korean broadcasting industry, short for eating show.)

First of all, the first example has a spacing and punctuation error. "붐 빕니다" shows inappropriate spacing results for the stem and the ending, and a comma is omitted from the list of consecutive items. Copy-bilinear had no improvement in error sentences. However, Filter shows the results of correction into natural sentences by inserting punctuation and unnecessary spacing.

In the second example, an error occurred when the word "A사" combined with Korean and English was recognized as a "의사" with a similar pronunciation. In addition, as punctuation is omitted, the readability of the input sentence is considerably lowered. Copy-bilinear does not solve the two problems, but rather shows the result of adding the spelling errors. On the other hand, Filter returns the result of removing all errors.

The third example is an awkward sentence that recognized "계" as "개." An error caused by the similar pronunciation of Korean significantly changes the context of the sentence. Copy-bilinear shows the result of partially solving the problem or changing it to a more awkward sentence. However, Filter shows the effect of fixing words that have been incorrectly translated from Korean to English and unnecessarily generated a weird word.

These results can indicate that errors in source text may appear as more than two problems, and the data-centric approach shows better performance than the model-centric approach in post-processing.

5. Conclusions

In this study, we investigated the data-centric and model-centric approaches based on BTS methodology [14] for the ASR post-processor. Furthermore, we also analyzed the negative effect and the results of the model-centric approach. We do not unconditionally believe in a particular approach and leverage the ASR post-processor to demonstrate the effectiveness of both approaches. However, we attempted to point out problems with the current trend of research pursuing a model-centric approach and warn against ignoring the importance of the data [30,31]. We hope to present why the data-centric approach should not be overlooked in deep-learning-based natural language processing research [32]. Based on that, we designed the experiments and proved the data-centric approach more effective than expected. For future work, we plan to build more data and conduct various verification for BTS based on hyperdata, and various verification will be conducted according to the amount of data.

Author Contributions: Funding acquisition, H.L.; investigation, C.P.; methodology, C.P.; project administration C.P.; conceptualization, C.P.; software, C.P. and J.S.; validation, C.P. and J.S.; formal analysis, C.P. and C.L.; writing—review and editing, C.P. and J.S. and S.L.; supervision, H.L. and I.A.; project administration, H.L.; data curation, C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Ministry of Science and ICT, Korea, under the Information Technology Research Center support program (IITP-2018-0-01405) supervised by the Institute for Information and Communications Technology Planning and Evaluation), and by the Basic Science Research Program through the National Research Foundation of Korea, funded by the Ministry of Education (NRF-2022R1A2C1007616)

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Many thanks to the KU NMT Group for taking the time to proofread this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Stuttle, M.N. A Gaussian Mixture Model Spectral Representation for Speech Recognition. Ph.D. Thesis, University of Cambridge, Cambridge, UK, 2003.
2. Gales, M.; Young, S. The application of hidden Markov models in speech recognition. *Found. Trends Signal Process.* **2008**, *1*, 195–304. [CrossRef]
3. Baevski, A.; Zhou, H.; Mohamed, A.; Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv* **2020**, arXiv:2006.11477.
4. Hjortnæs, N.; Partanen, N.; Rießler, M.; Tyers, F.M. The Relevance of the Source Language in Transfer Learning for ASR. In Proceedings of the Workshop on Computational Methods for Endangered Languages, Online, 2–3 March 2021 ; Volume 1, pp. 63–69.
5. Zhang, Z.Q.; Song, Y.; Wu, M.H.; Fang, X.; Dai, L.R. XLST: Cross-lingual Self-training to Learn Multilingual Representation for Low Resource Speech Recognition. *arXiv* **2021**, arXiv:2103.08207.
6. Ha, J.W.; Nam, K.; Kang, J.G.; Lee, S.W.; Yang, S.; Jung, H.; Kim, E.; Kim, H.; Kim, S.; Kim, H.A.; et al. ClovaCall: Korean goal-oriented dialog speech corpus for automatic speech recognition of contact centers. *arXiv* **2020**, arXiv:2004.09367.
7. Voll, K.; Atkins, S.; Forster, B. Improving the utility of speech recognition through error detection. *J. Digit. Imaging* **2008**, *21*, 371. [CrossRef]
8. Liao, J.; Eskimez, S.E.; Lu, L.; Shi, Y.; Gong, M.; Shou, L.; Qu, H.; Zeng, M. Improving readability for automatic speech recognition transcription. *arXiv* **2020**, arXiv:2004.04438.
9. Park, C.; Eo, S.; Moon, H.; Lim, H.S. 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, Virtual Event, 6–11 June 2021; pp. 97–104.
10. Wu, X.; Xiao, L.; Sun, Y.; Zhang, J.; Ma, T.; He, L. A survey of human-in-the-loop for machine learning. *Future Gener. Comput. Syst.* **2022**, *135*, 364–381. [CrossRef]

11. Roh, Y.; Heo, G.; Whang, S.E. A survey on data collection for machine learning: A big data-ai integration perspective. *IEEE Trans. Knowl. Data Eng.* **2019**, *33*, 1328–1347. [CrossRef]
12. Klein, G.; Zhang, D.; Chouteau, C.; Crego, J.M.; Senellart, J. Efficient and High-Quality Neural Machine Translation with OpenNMT. In Proceedings of the Fourth Workshop on Neural Generation and Translation, Virtual Event, 10 July 2020; pp. 211–217.
13. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
14. Park, C.; Seo, J.; Lee, S.; Lee, C.; Moon, H.; Eo, S.; Lim, H. 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), Bangkok, Thailand, 5–6 August 2021; pp. 106–116.
15. Koehn, P.; Chaudhary, V.; El-Kishky, A.; Goyal, N.; Chen, P.J.; Guzmán, F. Findings of the WMT 2020 shared task on parallel corpus filtering and alignment. In Proceedings of the Fifth Conference on Machine Translation, Virtual Event, 19–20 November 2020; pp. 726–742.
16. Gu, J.; Lu, Z.; Li, H.; Li, V.O. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv* **2016**, arXiv:1603.06393.
17. Luong, M.T.; Pham, H.; Manning, C.D. Effective approaches to attention-based neural machine translation. *arXiv* **2015**, arXiv:1508.04025.
18. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
19. Kudo, T.; Richardson, J. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv* **2018**, arXiv:1808.06226.
20. Polyzotis, N.; Zaharia, M. What can Data-Centric AI Learn from Data and ML Engineering? *arXiv* **2021**, arXiv:2112.06439.
21. Pan, I.; Mason, L.R.; Matar, O.K. Data-centric Engineering: Integrating simulation, machine learning and statistics. Challenges and opportunities. *Chem. Eng. Sci.* **2022**, *249*, 117271. [CrossRef]
22. Park, C.; Lim, H. A Study on the Performance Improvement of Machine Translation Using Public Korean-English Parallel Corpus. *J. Digit. Converg.* **2020**, *18*, 271–277.
23. Park, C.; Lee, Y.; Lee, C.; Lim, H. Quality, not Quantity? Effect of parallel corpus quantity and quality on Neural Machine Translation. In Proceedings of the 32st Annual Conference on Human & Cognitive Language Technology, Nice, France, 25–29 October 2020 ; pp. 363–368.
24. Gale, W.A.; Church, K. A program for aligning sentences in bilingual corpora. *Comput. Linguist.* **1993**, *19*, 75–102.
25. Napoles, C.; Sakaguchi, K.; Post, M.; Tetreault, J. 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, Beijing, China, 26–31 July 2015; Volume 2: Short Papers, pp. 588–593.
26. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: A method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics, Philadelphia, PA, USA, 7–12 July 2002; pp. 311–318.
27. Choi, J.M.; Kim, J.D.; Park, C.Y.; Kim, Y.S. Automatic Word Spacing of Korean Using Syllable and Morpheme. *Appl. Sci.* **2021**, *11*, 626. [CrossRef]
28. Yi, J.; Tao, J.; Bai, Y.; Tian, Z.; Fan, C. Adversarial transfer learning for punctuation restoration. *arXiv* **2020**, arXiv:2004.00248.
29. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016 ; pp. 2383–2392.
30. Seo, J.; Lee, S.; Park, C.; Jang, Y.; Moon, H.; Eo, S.; Koo, S.; Lim, H.S. A Dog Is Passing Over The Jet? A Text-Generation Dataset for Korean Commonsense Reasoning and Evaluation. In Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2022, Virtual Event, 10–15 July 2022; pp. 2233–2249.
31. Kang, M.; Seo, J.; Park, C.; Lim, H. Utilization Strategy of User Engagements in Korean Fake News Detection. *IEEE Access* **2022**, *10*, 79516–79525. [CrossRef]
32. Ranaldi, L.; Fallucchi, F.; Zanzotto, F.M. Dis-Cover AI Minds to Preserve Human Knowledge. *Future Internet* **2021**, *14*, 10. [CrossRef]