Korean "Airbnb" Script Deobfuscation Using Sequence-to-Sequence Models.

Abdul Rafay
Boston University
rafaya@bu.edu

Mehmet Bora Sarioglu
Boston University
Sarioglu@bu.edu

Abstract

Korean online communities have developed a unique form of phonetic text coding for discreet communication. This project aims to develop an automated decoder for this coded Korean text. We will create a dataset of coded and decoded text, train a model to recognize transformation patterns, and deploy a functional tool for real-time decoding. This tool will bridge the communication gap, improving accessibility and understanding of online discourse.

1 Introduction

1.1 Problem Statement

Koreans online often use a phonetic text coding system popularly known as "airbnb script" to communicate candidly, sometimes obscuring messages. By subtly altering word phonetics, users convey meaning while maintaining a degree of privacy. This practice is common in online reviews, social media, and forums. While effective for native speakers, this coded language presents challenges for automated systems and non-Koreans.

1.2 Objective

Our goal for this project is to experiment and develop a decoder tool of the "aribnb script". The tool features a simple interface where users input encoded Korean text and the decoded standard Korean is displayed. The tool will provide real-time decoding with immediate feedback as users type or paste text.

1.3 Potential Impact

This project may have significant potential impact on facilitating understanding of Korean online discourse. It may also provide a valuable resource for studying Korean text phonetic transformations and at the core will offer a solution to a niche problem allowing for further applications. For businesses, this tool will enable better understanding of customer feedback that may contain encoded Korean text.

1.4 About Text Obfuscation

Text obfuscation—the deliberate alteration of text to make it difficult for automated systems to process while remaining comprehensible to humans—poses significant challenges across multiple domains. Although we are interested in specific case of Korean "airbnb script", for better understanding of the concept, we can try to understand it through the context of English first. Common examples include replacing characters with visually similar ones (e.g., "noon" with "nouan"), substituting words with phonetically similar alternatives (e.g., "meet me later" $\rightarrow$ "meat mi letter"), or other transformations that preserve human readability while confusing automated systems.

These obfuscation techniques are frequently employed to:

- Evade content moderation systems
- Bypass plagiarism detection
- Circumvent keyword filtering
- Avoid automated content analysis

```
Original: Please meet me at the coffee shop
tomorrow at noon.
Obfuscated: Pleezi meat mie it tha koffee shop
timariw et noun.
```

Figure 1: Example of obfuscation.

Our research aims to develop a robust system for deobfuscation—converting obfuscated text back to its standard form—by leveraging phonetic representations as an intermediate step. This approach is motivated by the observation that many obfuscation techniques maintain phonetic similarity to preserve human readability.

가각간갇갈갉갊감갑값갓갔강갖갗같갚갛개객갠갤갬갭갯갰갱갸갹갼걀걋걍거걱건걷걸걺검겁것겄겅겆겉겊겋게겐
겔겜겝겟겠겡겨격겪견결겸겹겻겼경곁계곈곌곕곗고곡곤곧골곪곬곯곰곱곳공곶과곽관괄괆괌괍괏광괘괜괠괩괬괭
괴괵괸괼굄굅굇굉교굔굘굡굣구국군굳굴굵굶굻굼굽굿궁궂궈궉궐궜궝궤궷귀귁귄귈귐귑귓규균귤그극근귿글긁금
급긋긍긔기긱긴긷길긺김깁깃깅깆깊까깍깎깐깔깖깜깝깟깠깡깥깨깩깬깰깸깹깻깼깽꺄꺅꺌꺼꺽꺾껀껄껌껍껏껐껑
께껙껜껨껫껭껴껸껼꼇꼈꼍꼐꼬꼭꼰꼲꼴꼼꼽꼿꽁꽂꽃꽈꽉꽐꽜꽝꽤꽥꽹꾀꾄꾈꾐꾑꾕꾜꾸꾹꾼꿀꿇꿈꿉꿋꿍꿎꿔
꿜꿨꿩꿰꿱꿴꿸뀀뀁뀄뀌뀐뀔뀜뀝뀨끄끅끈끊끌끎끔끕끗끙끼끽낀낄낌낍낏낑나낙낚난날낡낢남납낫났낭낮낯낱낳내
낵낸낼냄냅냇냈냉냐냑냔냘냠냥너넉넋넌널넓넘넙넛넜넝넣네넥넨넬넴넵넷넸넹녀녁년념녑녔녕녘녜노녹논놀놂놈
놉놋농높놓놔놘놜놨뇌뇐뇔뇜뇝뇟뇨누눅눈눋눌눔눕눗눙눠눴눼뉘뉜뉠뉨뉩뉴뉵뉼늄늅늉느늑는늘늙늚늠늡늣능늦
늪늬늰늴니닉닌닐닒님닙닛닝닢다닥닫달닭닮닯담답닷닸당닺닻닿대댁댄댈댐댑댓댔댕댜더덕던덛덜덤덥덧덩덫
덮데덱덴델뎀뎁뎃뎄뎅뎌뎐뎔뎠뎡뎨도독돈돋돌돎돐돔돕돗동돛돝돠돤돨돼됐되된될됨됩됫되된될됨됩됴두둑둔둘둠
둡둣둥둬뒀뒈뒝뒤뒨뒬뒵뒷뒹듀든듣들듦듬듭듯등듸디딕딘딛딜딤딥딧딨딩따딱딴딸딤...

Figure 2: All 11,172 valid Hangul syllables (가–) as continuous text.

2 Existing Related Work

There aren't any prexisting baseline method to our niche problem, but we were inspired by following relevant works from the field in developing our solution:

There aren't any preexisting baseline methods for our niche problem, but we were inspired by the following relevant works from the field in developing our solution:

2.1 Text Normalization

Previous approaches to text normalization have primarily focused on handling social media text, where abbreviations and non-standard spellings are common. Limsopatham and Collier (1) adapted phrase-based machine translation techniques to normalize medical terminology in social media messages, demonstrating the effectiveness of treating text normalization as a translation problem. Similarly, Bojja et al. (2) explored using machine translation frameworks with incentivized feedback mechanisms for normalizing social media text in mobile game contexts.

2.2 Adversarial Text Detection

Work on detecting adversarial text modifications includes methods by Kumarage et al. (4), who developed assessment frameworks for evaluating AI-generated text detectors and explored evasion techniques using soft prompts. However, these methods focus primarily on detection rather than recovery of the original text, which is a key distinction in our approach to Korean text deobfuscation.

2.3 Phonetic Modeling in NLP

Phonetic representations have been leveraged in various NLP tasks. Honnet et al. (5) developed strategies for normalizing Swiss German dialects using machine translation approaches that account for phonetic variations, which bears similarities to our Korean deobfuscation task. Furthermore, Tiwari and Naskar (6) applied deep neural networks specifically for social media text normalization, showing how modern neural architectures can effectively learn the complex patterns in non-standard text. Our work extends these approaches to the specific task of Korean text deobfuscation, leveraging phonetic similarities between obfuscated and standard forms.

2.4 Traditional Approaches

There being no relevant existing baseline models, we devised our own approach by exploring following traditional approaches to text normalization in detail:

2.4.1 Rule Based Systems

Our initial approach to the problem was to figure out character-level transformations rules from the obfuscation tool and potentially reverse it. However, we learned that the relationship between obfuscated text and standard Korean text was many to one, and we needed a more complex methodology that can learn a huge amount of variations. Korean characters have total of 11,172 possible syllable combinations as seen in Figure 2, which makes it very complex to defined rules for standard to obfuscated text. Thus, rule-based systems as a sole approach was not feasible.

2.4.2 Statistical n-gram models

This approach was attractive in the sense that it could account for the huge amount of various combinations being trained on large corpora. We knew that the obfuscations were happening on sub-character level phonetic transformations, so we created our n-gram model to work on Jamo components of each Hangul character. Even though it was a promising idea the complexity of the obfuscations and the lack of depth in terms of context of our n-gram model it performed poorly.

2.4.3 Neural Machine Translation

Treating obfuscated text as a "source language" to be translated and treating the original text as the "target language" we trained character level and sub-character level LSTM models. They were significantly stronger than the statistical models but still came short of providing a useful degree of de-obfuscation.

3 Method

Our final approach combines phonetic preprocessing of the obfuscated text in order to mimic human though process or behavior with a character level sequence-to-sequence model to recover original text from obfuscated versions.

3.1 Phonetic Preprocessing

How can Koreans understand encoded text, without non native speaker knowing? Answer to this was the pronunciation. Koreans automatically pronounce those encoded text in their head which happens to sound similar to regular phrases. From this realization, we tried to look for ways to convert encoded text into phonetic representations that could be easily parsed or trained.

3.1.1 g2pK Module

Hangul, the main script for Korean, is phonetic, but the pronunciation rules are notoriously complicated. That's why we employed the g2pK (Grapheme-to-Phoneme) library specialized for Korean (available at https://github.com/Kyubyong/g2pK) to convert both original and obfuscated texts into phonetic representations that matches the actual sound of the phrases. This conversion maps visually distinct but phonetically similar characters and character sequences to the nearly similar phonetic representation, providing our model with a more consistent input space.

```
Original: 어제는 날씨가 맑았는데...
Obfuscated: 얻쩨눈 냘쒸카 맑앗눈뎨...
g2pK conversion: 언쩨눈 냘쒸카 말간눈뎨...
```

Figure 3: Example of g2pK Conversion.

So, this would convert obfuscated text to a phonetic representation that captures their sound patterns, regardless of the specific characters used to create those sounds, making it more similar to original text.

3.1.2 Hangul Romanization

Another way of phonetic representation explored for potential integration into pipeline was romanization of Korean alphabets. There was a python module (available at https://github.com/osori/korean-romanizer) that comply with the rule developed by the National Institute of Korean Language, the official romanization system being used in the Republic of Korea. However, we ended up using g2pK only.

3.2 Character-Level Tokenization

Rather than using subword or word-level tokenization, we implement a character-level tokenizer that treats each character as a distinct token. Our tokenizer includes special tokens for padding (<pad>), unknown characters (<unk>), sequence start (<s>), and sequence end (</s>).

3.3 Model Architeture

We employ a BART-based (Lewis et al., 2020) sequence-to-sequence architecture with the following specifications:

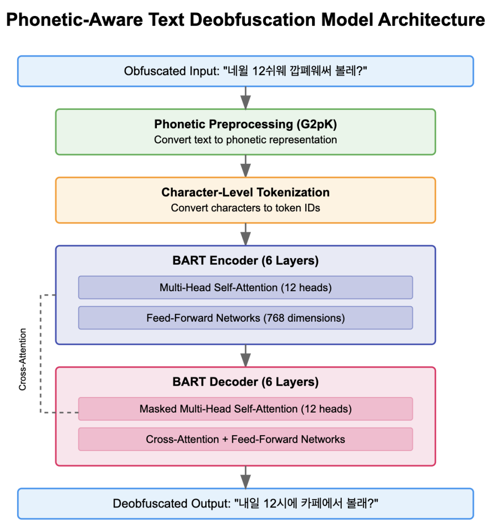


Figure 4: Model Architecture Diagram

- Vocabulary size based on unique characters in our dataset
- 6 encoder layers and 6 decoder layers
- 768-dimensional hidden representations
- 12 attention heads in both encoder and decoder

- Maximum position embeddings of 1024 tokens

The model inputs are the phonetic representations of obfuscated text, and it is trained to output the phonetic representations of the corresponding original text. In addition to this we have also tried to implement this pipeline using KoBART a pretrained model specialized in the context of Korean language. However, we focused on base BART with our configurations to it. KoBART was promising and had better metrics compared to base BART with small data, however the lack of time and the inconsistency of the SCC cluster resulted in us not being able to train it on a large dataset, therefore we decided not to use it as our final model.

3.4 Training Procedure

We trained our model using cross-entropy loss with the following hyperparameters:

- Learning rate: 3e-5
- Batch size: 16 per device
- Weight decay: 0.01
- Training epochs: 10
- FP16 precision (when GPU is available)

Training was done on SCC, and took 14 hours to complete.

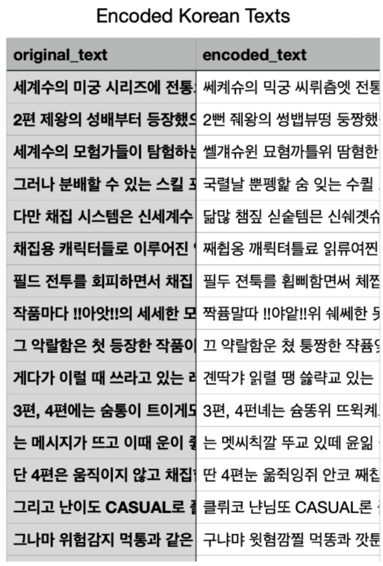
4 Experiments

4.1 Dataset

We created a large-scale dataset of paired original and obfuscated texts. For this we borrowed text data from **KoWiki** dataset, an Korean Wikipedia counterpart. It consists of the headers and text of Wikipedia in Korean. To generate the obfuscated dataset, we had to create the obfuscator ourselves, before we could train our model. We scraped the the most popular tool by hanmesoft (available at https://airbnbfy.hanmesoft.com/) and retrieved the javascript file with the obfuscation logic that was being used in the frontend of the website and used the same code to write our own obfuscation code. The dataset contains approximately 10 million text pairs, of which we used fraction for training, with each pair consisting of an original text and its obfuscated version. The obfuscation includes character substitutions, phonetic spelling variations, and other transformations that preserve the general phonetic structure.

Dataset	Lines	Size
Train	26,794,425	1.7G
Dev	130,419	7.7M
Test	134,340	8.4M

Table 1: KoWiki Dataset Information


Encoded Korean Texts

original_text	encoded_text
세계수의 미궁 시리즈에 전통	쎄케슈의 믹궁 씨뤼츰엣 전통
2편 제왕의 성배부터 등장했	2뻔 줴왕의 썽뱁뷰떵 둥짱했
세계수의 모험가들이 탐험하	쎌개슈윈 묘험까틀위 땀혐한
그러나 분배할 수 있는 스킬	국렬날 뿐펭핥 숨 잊는 수퀼
다만 채집 시스템은 신세계수	닮많 챕짚 싣숱템믄 신쉐겟슈
채집용 캐릭터들로 이루어진	째췹옹 깨뤽터틀료 읽류여찐
필드 전투를 회피하면서 채집	필두 전툭를 휩삐함면써 체쩝
작품마다 !!아앗!!의 세세한 모	짝품말따 !!야앝!!위 쉐쎄한
그 악랄함은 첫 등장한 작품이	끄 약랄함운 쳤 퉁짱한 쟉품
게다가 이럴 때 쓰라고 있는	곈딱갸 읽렬 땡 쓿략교 있는
3편, 4편에는 숨통이 트이게	3편, 4펀녜는 슘똥위 뜨윅케
는 메시지가 뜨고 이때 운이	는 멧씨칙깔 뚜교 있떼 윤잂
단 4편은 움직이지 않고 채집	딴 4편눈 욺줙잉쥐 안코 째칩
그리고 난이도 CASUAL로	클뤼코 냔님또 CASUAL론
그나마 위험감지 먹통과 같은	구냐먀 윗혐깜찔 먹똥콰 깟튼

Figure 5: Example Pairs from Dataset

4.2 Evaluation Metrics

We evaluate our model using the following metrics:

Character Error Rate (CER): The Levenshtein distance between predicted and original texts, normalized by the length of the original text.

Word Error Rate (WER): The WER is derived from the Levenshtein distance, working at the word level instead of the phoneme level.

BLEU Score: To measure the general similarity between predicted and original texts.

4.3 Baselines

Before settling with our phonetic-aware approach, we tried following baselines:

- **LSTM:** We set up a character-level sequence to sequence where we created our of Encoder and Decoder architecture with LSTMs that we wrote ourself using pytorch.

- **Rule-based Normalization:** A system using handcrafted rules for common obfuscation patterns (e.g. "ㅈ" → "ㅊ", "ㅂ" → "ㅃ").
- **Character-level JAMO Transformer:** We used character-level tokenization on decomposed Hangul characters (Jamos) to enhance phonetic representation. We also introduced Jamo embeddings to improve context understanding. Special tokens handle sequence padding and start/end markers.

4.4 Results

Our phonetic-aware model outperforms all baselines, with a particularly significant improvement in BLEU score . We also did poll with native Korean speakers on our model's performance. The feedback we got implied that the model was doing a good job in general, on a scale out of 10 we got an average around 8 in terms of deobfuscating 'Korean AirBnB Script'. Moreover, the native Korean speakers who tried the website said that model struggled slightly with foreign names and domain spesific words. From the feedback we got the model may benefit from more training data for non-korean words written with Hangul characters.

Model Variant	WER	CER	BLEU
Latest Model	**0.013**	**0.393**	**0.82**
LSTM	-	0.23	0.17

Table 2: The huge performance increase we got fine tuning transformers

4.5 Error Analysis

Based on common errors made by our model we found out that:

1. The model struggles with severe obfuscations.
2. Foreign noun based hangul are error prone.
3. Domain-specific terms and proper nouns are more challenging to deobfuscate correctly.

5 Conclusions and Future Work

Our research demonstrates that a phonetic-aware sequence-to-sequence approach significantly improves text deobfuscation performance compared to traditional methods in the case of Korean "airbnb script". The use of phonetic representations helps bridge the gap between visually distinct but phonetically similar text, allowing our model to more effectively recover original text from obfuscated versions. Key findings from our work include:

- Phonetic preprocessing provides substantial benefits for text deobfuscation tasks.
- Character-level modeling is more effective than word-level approaches for this problem.
- Modern transformer architectures can effectively learn the complex mappings between obfuscated and original text forms.

Future work directions include:

1. Train on a even bigger dataset. We believe that this will be the most significant way to improve the performance of the model to be production ready.
2. Extending the approach to handle multiple languages and multilingual obfuscation. Potentially other language that uses phonetic alphabets, to incorporate G2P approach. This may also lead to a to foreign noun error we found with our models.
3. Incorporating visual similarity features to better handle cases where characters are substituted based on visual rather than phonetic similarity.
4. Developing adaptive models that can handle novel obfuscation techniques not seen during training.
5. Exploring applications in content moderation systems and accessibility tools. Potentially also assessing our model's impact if any in the Korean online discourse after refinement and web deployment.

Our work contributes to the growing body of research on text normalization, for a very niche case of Korean text deobfuscation and provides practical tools for applications in content moderation, security, and accessibility.

6 Replicability

All the models and data used in our experiments are available at our GitHub repository: https://github.com/Abdul03Rafay/KoreanDeobfuscator.

The repository includes:

- Preprocessing scripts for phonetic conversion

- Model implementation using HuggingFace Transformers
- Training scripts
- Sample data (full data was to large to be uploaded.)
- Front-end funtionality.

For full reproducibility, we also provide trained model checkpoints and detailed documentation on the environment setup.

7 Web Application

We've developed an website that bridges the communication gap for non-Korean speakers encountering obfuscated Korean text. Our intuitive three-step platform first allows users to experiment with Korean text obfuscation, giving them insight into how this process works. The core functionality lies in our transformer model that we have trained which deobfuscate 'Korean AirBnB Script'. For non-Korean users, we've integrated the Helsinki translation model, enabling translation of the deobfuscated text. This way we can help users not only decode obfuscated Korean text but also understand its meaning, making digital Korean content more accessible regardless of obfuscation techniques.

References

[1] Limsopatham, N., & Collier, N. (2015). Adapting Phrase-based Machine Translation to Normalise Medical Terms in Social Media Messages. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1675-1680.

[2] Bojja, N., Nedunchezhian, A., & Wang, P. (2015). Machine translation in mobile games: augmenting social media text normalization with incentivized feedback. In *Proceedings of Machine Translation Summit XV: User Track*.

[3] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871-7880.

[4] Kumarage, T., Sheth, P., Moraffah, R., Garland, J., & Liu, H. (2023). How Reliable Are AI-Generated-Text Detectors? An Assessment Framework Using Evasive Soft Prompts. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

[5] Honnet, P. E., Popescu-Belis, A., Musat, C., & Baeriswyl, M. (2018). Machine Translation of Low-Resource Spoken Dialects: Strategies for Normalizing Swiss German. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

[6] Tiwari, A. S., & Naskar, S. K. (2017). Normalization of Social Media Text using Deep Neural Networks. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*.

Figure 6: Bad Example With Partially Wrong Deobfuscation

Figure 7: Good Example With Perfect Deobfuscation.