UKoSpeech: A Universal Korean ASR System for Diverse Domains

Anonymous ACL submission

Abstract

The rapid advancement of Automatic Speech Recognition (ASR) systems has dramatically transformed transcription processes, minimizing the need for expert human intervention. Despite the growth in ASR technologies and the emergence of robust models like Whisper, significant challenges remain. Specifically, the scarcity of non-English training data and poor adaptability to domain-specific contexts hinder broader application. This paper introduces UKoSpeech, a novel Korean ASR system designed to address these issues through a unique two-pronged approach: a Korean data curation pipeline leveraging domain-specific data from sources such as YouTube subtitles, and a domain-specific training framework that utilizes a domain prompt technique for enhanced adaptability. Our results indicate that UKoSpeech not only fills the gap in multilingual ASR research but also provides superior domain-specific performance compared to established ASR systems like Whisper, Google STT, and CLOVA Speech. Through extensive evaluation across diverse domains such as finance, medicine, and law, UKoSpeech demonstrates state-of-the-art performance, establishing a new benchmark for domain-adaptable ASR systems.

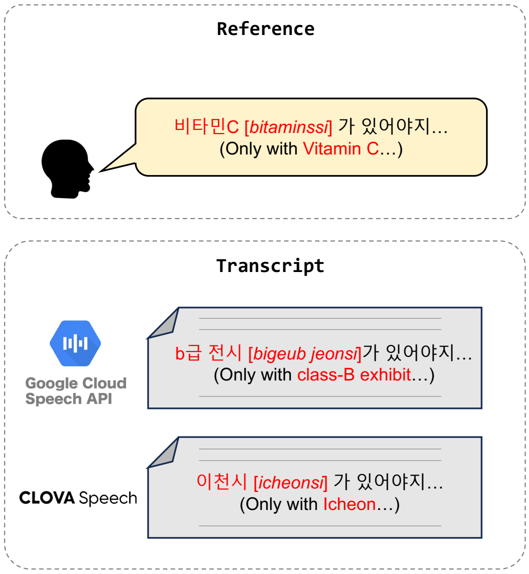

Figure 1: Example of transcript generated by Google Speech API and ClOVA Speech. Both ASR systems exhibited errors in transcribing the medical term "Vitamin C" (colored red in the transcript). The pronunciation of each vocabulary is marked with *Italic* font.

1 Introduction

The evolution of Automatic Speech Recognition (ASR) has significantly reduced the reliance on expert human labor for transcription tasks. Initially, the prospect of converting speech to text through computational means was intriguing, but it was blocked by high error rates that precluded practical application (Li et al.). As research advanced, transcription accuracy improved markedly (Dhanjal and Singh, 2023), leading to the widespread deployment of ASR-based commercial systems in real-world scenarios.

In recent years, there are numerous amount of publicly available ASR models (Pratap et al., 2020; Povey et al., 2011; Radford et al., 2023; Baevski et al., 2020). Among these, Whisper (Radford et al., 2023) is a prominent ASR model that is pre-trained with large amounts of paired audio-text multilingual data. Thorough experiments reveal that Whisper demonstrates superior robustness in handling noise backgrounds compared to existing ASR system (Radford et al., 2023).

Despite the promising utility of these ASR systems, such as Whisper, we witness two primary limitations. First, the amount of existing non-English ASR training data is limited. While research on improving multilingual performance of ASR is thriving, it is undeniable that the majority of studies are conducted in English. This inevitably leads to resource constraints of languages other than English. As a result, non-English data remains relatively scarce in the ASR community (Yang et al., 2023;

Bartelds et al., 2023).

Second, the low adaptability to specific domain is notable. As illustrated in Figure 1, popular commercial ASR systems such as Google STT[1](Speech-to-Text) and CLOVA Speech[2] exhibit poor performance in Korean domain-specific speech. As highlighted in (Nanayakkara et al., 2022), lack of adaptability to specific domains is critical as even the subtle transcription errors could potentially modify the meaning of transcript, making them unusable in real-world applications.

To address these challenges, we propose **UKoSpeech**, the Korean ASR that is universally proficient across multiple domains. UKoSpeech is developed with **(1)** paired audio-text Korean dataset collected through data curation pipeline and **(2)** is trained with domain-aware ASR training framework. We experiment with Korean, a morphologically-rich language in which ASR systems struggle to generate accurate results (Park et al., 2021, 2024).

Specifically, our data curation pipeline enhances existing methodologies (Lakomkin et al., 2018) by leveraging subtitles available on YouTube. In other words, our framework enables collection of any Korean data from specific domains, thus providing solution to building reliable and quality datasets.

Inspired from (Liao et al., 2023), our domain-aware ASR training framework integrates domain prompt to effectively orient the model toward specific domain. It enables model to adapt its proficiency in transcribing audio from various domains simply by changing the domain prompt.

In our experimental evaluation, we showcase that UKoSpeech achieves state-of-the-art performance across domains such as finance, medicine, and law. Notably, it surpasses base Whisper model, CLOVA Speech and Google STT, showing the strong domain adaptation capability of our proposed framework.

2 Related Works

Adapting ASR models to specific domains has been a significant focus within the ASR community. Domain adaptation strategies fall into three main categories. The first involves modifying the decoding strategy to enhance the recognition of predetermined domain-specific vocabularies by focusing on contextual information (Jung et al., 2022; Kocour et al., 2021; Zhao et al., 2019). The second category involves appending a 'domain prompt' to the decoder. In this approach, a small set of domain embedding parameters are trained within an auxiliary language model, and the output from this model informs the generation of the next word token in the ASR model (Dingliwa et al., 2022). Similarly, (Liao et al., 2023) append prompts containing domain-related tags during the decoding process. The third strategy increases the training dataset size by synthesizing audio from text using a text-to-speech (TTS) model (Joshi and Singh, 2022; Vásquez-Correa et al., 2023), which is then used for ASR training.

However, each method has its limitations. The first strategy lacks flexibility across diverse languages due to the unique structural characteristics of each language(Koplenig et al., 2023), necessitating distinct decoding strategies for different languages. As our work does not rely on language-specific decoding strategies, it holds potential for adaptation to other languages, though it currently focuses on developing a Korean ASR model. Our approach aligns most closely with the second category, but we do not utilize larger language model such as (Dingliwa et al., 2022), and our model utilizes single domain tag, rather than multiple domain tag represented in (Liao et al., 2023). This adaptation enhances the overall practicality of our model. The third strategy is dependent on the quality of TTS model, with potential for suboptimal ASR performance when trained on such synthesized data. Our methodology employs authentic audio paired with curated human-annotated text, ensuring both robustness and reliability in the model.

3 Data Curation Pipeline

ASR Dataset tailored for Korean, such as ZerothKorean[3], KsponSpeech (Bang et al., 2020), Korean university lecture recordings[4], emotion-tagged free conversation data[5], anchor voice dataset[6] offer potential augmentation of over 10,000 hours to the existing corpus. Nonetheless, these corpus still **(1)**

[1]https://cloud.google.com/speech-to-text

[2]https://clova.ai/speech

[3]https://huggingface.co/datasets/Bingsu/zeroth-korean

[4]https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=data&dataSetSn=71627

[5]https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=data&dataSetSn=71631

[6]https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=data&dataSetSn=71557

Algorithm 1 Crawling raw data from YouTube

```
Parameters: Keyword set K
D ← ∅                                    ▷ Initialize empty output set
i ← 0                                    ▷ Initialize index of crawled data
for k_j ∈ K do
    C ← EXTRACTCHANNELNAMES(k_j)
    for c_k ∈ C do
        V ← EXTRACTVIDEOIDS(c_k)
        for v_l ∈ V do
            if NOT CHECKMACHINEGEN(v_l) then
                a_i ← EXTRACTAUDIO(v_l)
                S_i, E_i ← EXTRACTTIMESTAMP(v_l)
                T_i ← EXTRACTTRANSCRIPT(v_l)
                D ← D ∪ {a_i, S_i, E_i, T_i}
                i ← i + 1
            end if
        end for
    end for
end for
```

lack explicit domain annotations **(2)** confined to particular domains **(3)** grouped into subjects rather than domains.

To address this issue, we introduce a data curation pipeline designed to construct Korean domain-specific dataset for ASR training. This pipeline operates in two consecutive phases: the initial phase involves the extraction of audio and corresponding transcript from YouTube while the subsequent phase focuses on prepossessing this raw data into formats suitable for training and testing.

3.1 First phase: Crawling raw data from YouTube

Selecting an appropriate source for crawling is crucial to obtain reliable and high-quality data. Given its rich content encompassing both audio and text, YouTube has attracted many researchers to use it as a source for data acquisition (Lakomkin et al., 2018; Chen et al., 2021; Zhang et al., 2023; Takamichi et al., 2021), thus we have selected it as our crawling source. The crawling process operates in the following manner:

1. We determine the keywords to search on YouTube.

2. Our crawler extracts channels relevant to the keywords and acquire unique video IDs of all videos uploaded by those channels. Note that only the videos that do not trespass legal issues are selected during this process. Details regarding legal issues are discussed in ethics statements.

3. Our crawler gathers audio, timestamp information and transcripts from videos. Given the potential for transcripts to be auto-generated by machine, crawler assess their metadata through YouTubeTranscriptApi class implemented in youtube_transcript_api[7] python library. This metadata includes details on whether humans or machines created the transcripts, guiding our crawler to collect audio and transcripts only after verifying human authorship.

We set initial keywords as '금융 유튜버' (Financial YouTuber), '의학 유튜버' (Medical YouTuber) and '법률 유튜버' (Legal YouTuber), in order to build dataset that consists of finance, medical and law domains, respectively.

We define the dataset acquired from the initial phase as $D = \bigcup_{i=1}^{N} \{a_i, S_i, E_i, T_i\}$, where a_i denotes the i-th audio, and S_i and E_i denote the sequences of start and end times of the audio segment, respectively, acquired from timestamp information. T_i denotes the sequence of transcripts, and N denotes the total number of extracted videos. The sequences S_i, E_i, and T_i are defined as follows:

$$\begin{aligned} S_i &= [s_i^1, s_i^2, \ldots, s_i^j, \ldots, s_i^{M_i}] \\ E_i &= [e_i^1, e_i^2, \ldots, e_i^j, \ldots, e_i^{M_i}] \\ T_i &= [t_i^1, t_i^2, \ldots, t_i^j, \ldots, t_i^{M_i}] \end{aligned} \tag{1}$$

M_i represents the total number of transcripts for the i-th audio. Therefore, t_i^j is the transcript of the i-th audio corresponding to the timestamp from s_i^j to e_i^j. Algorithm 1 summarizes the overall process in the first phase.

3.2 Second phase: Prepossessing raw data

The typical way of creating a training dataset for ASR would be segmenting each audio a_i by s_i^j and e_i^j, resulting in pairs of segmented audio and its corresponding transcript t_i^j. However, the duration of the segmented audio, $e_j - s_j$ may be either excessively short or long. Training on short-sized audio segments can degrade performance (Press et al., 2021), while training on longer segments may be infeasible due to limited context size of ASR systems. This led us to employ a preprocessing technique to create data that are close to maximum context size of ASR, but not exceeding it. For sequence of transcript text T_i, we concatenate the transcripts inside sequence $[t_i^j, \ldots, t_i^k]$ to create a single concatenated transcript t_{concat}, where t_{concat}

[7] https://github.com/jdepoix/youtube-transcript-api

meets the condition $(e_i^k - s_i^j) \leq \theta$ and θ denotes maximum context size of ASR, counted in seconds. Since the baseline model of UKoSpeech is Whisper, we select a value of θ to be maximum context size of Whisper, which is 30.

We then segment the audio based s_i^j and e_i^k to create segmented audio a_{seg}.

Despite successfully preprocessing crawled transcripts, verifying their authenticity as accurate representations of the speech within audio remains challenging. To address this, we implement a filtering process comprising three sequential steps: **(1)** ASR model $\mathcal{M}$ is employed to generate transcripts of segmented audio, serving as a reference against crawled transcripts. For clarity, we denote the ASR-generated transcript as the **reference text** $t_{\text{ref}} = \mathcal{M}(a_{\text{seg}})$ and crawled transcript as **hypotheses text** $t_{\text{hyp}} = t_{\text{concat}}$. **(2)** Normalize both the reference and hypotheses texts, followed by the computation of the error rate between them. Normalization is performed to facilitate a fair comparison between two input texts. **(3)** Filter out hypotheses texts that exceed a predetermined error rate threshold δ, thereby ensuring the selection of texts that faithfully capture the audio content. In summary, we select t_{hyp} that meets the following criteria:

$$f_e(f_n(t_{\text{ref}}), f_n(t_{\text{hyp}})) < \delta \tag{2}$$

where $f_n(\cdot)$ denotes normalize function, $f_e(\cdot)$ denotes error rate function.

We employ WhisperX-large-v2(Bain et al., 2023), a faster whisper model than its predecessors, Whisper-large-v2, as model for generating reference texts. For f_e, we use Character Error Rate (CER), as it is more reliable than WER since Korean is a syllabic and character-based language. For f_n, we use KoreanNormalizer function that is implemented in KoLM[8] Python library. It normalizes text by following process:

- Remove all special characters and surrounding whitespaces.
- Modify all Arabic numerals to Korean letter.
- Modify all alphabets to Korean letter.

We select a δ value of 5.31 to include approximately 25% of the entire dataset, resulting in a dataset totaling around 12 hours in duration.

Lastly, we identified and corrected instances where periods were inappropriately placed at the end of sentences using the Kiwipiepy library[9], which allowed us to decompose t_{hyp} into morpheme-level units and append periods where necessary.

The preprocessing techniques described are performed for all i to acquire the preprocessed dataset $\tilde{D} = \bigcup_{k=1}^{N'} a_{\text{seg}}^k, t_{\text{hyp}}^k$, where N' denotes the total number of concatenated audio and transcript pairs. The entire process is detailed in Algorithm 2.

Algorithm 2 Prepossessing of Raw Data

Required: A dataset consisting of audios, sequence of start time of audio, end time of audio and its corresponding transcript $D = \bigcup_{i=1}^{N}\{a_i, S_i, E_i, T_i\}$, where $S_i = [s_i^1, s_i^2, \ldots, s_i^j, \ldots, s_i^{M_i}]$, $E_i = [e_i^1, e_i^2, \ldots, e_i^j, \ldots, e_i^{M_i}]$, $t_i = [t_i^1, t_i^2, \ldots, t_i^j, \ldots, t_i^{M_i}]$

Parameters: Maximum context size of ASR θ, ASR model $\mathcal{M}$, Error rate threshold δ, Normalize function $f_n(\cdot)$, Error rate function $f_e(\cdot)$

$\tilde{D} \leftarrow \emptyset$ ▷ Initialize empty output set
for $a_i, S_i, E_i, T_i \in D$ **do**
  $\tau \leftarrow 0$ ▷ Initialize duration of concatenated audio
  $k \leftarrow 0$ ▷ Initialize index of concatenated audio and transcript
  $\ell \leftarrow 0$ ▷ Initialize left index
  $r \leftarrow 0$ ▷ Initialize right index
  while $r < M_i$ **do**
    $\Delta \leftarrow e_i^r - s_i^l$
    if $\tau + \Delta \leq \theta$ **then**
      $\tau \leftarrow \tau + \Delta$
    else
      $t_{\text{concat}}^k \leftarrow$ ""
      for $t_{\text{temp}} \in [t_i^l, \ldots, t_i^{r-1}]$ **do**
        $t_{\text{concat}}^k \leftarrow t_{\text{concat}}^k + t_{\text{temp}}$
      end for
      $a_{\text{seg}}^k \leftarrow \textsc{SegmentAudio}(a_i, s_i^l, e_j^{r-1})$
      $t_{\text{ref}}^k \leftarrow \mathcal{M}(a_{\text{seg}}^k)$
      if $f_e(f_n(t_{\text{concat}}^k), f_n(t_{\text{ref}}^k)) < \delta$ **then**
        $t_{\text{concat}}^k \leftarrow \textsc{AppendPeriod}(t_{\text{concat}}^k)$
        $\tilde{D} \leftarrow \tilde{D} \cup \{a_{\text{seg}}^k, t_{\text{concat}}^k\}$
        $k \leftarrow k + 1$
      end if
      $\tau \leftarrow e_i^r - s_i^r$
      $\ell \leftarrow r$
    end if
    $r \leftarrow r + 1$
  end while
end for

4 Domain-Aware ASR Tuning

Since Whisper is trained with audio data segmented with 30 seconds in duration, it receives previous text tokens for long-term transcription. The previous text tokens are placed between `|startofprevious|` and `<|startoftranscript|>` special tokens in

[8] https://github.com/scarletcho/KoLM

[9] https://github.com/bab2min/kiwipiepy

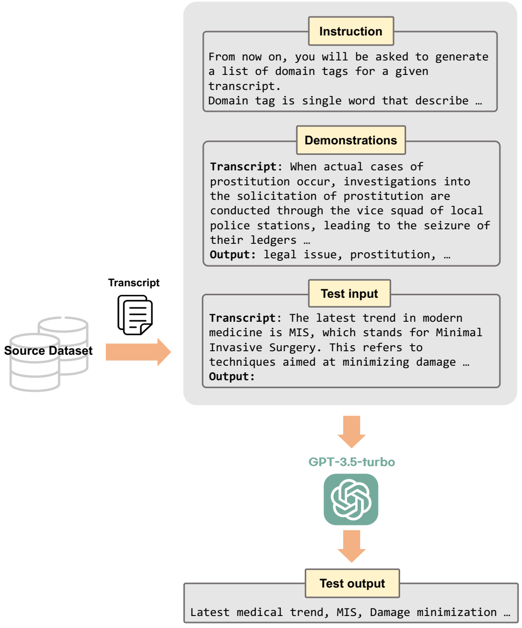

Figure 2: Domain tag generation with GPT-3.5-turbo. The input prompt comprises instructions, several demonstrations, and a transcript.

decoding stage. Whisper then autoregressively generates <|language|> and <|transcribe|> tokens followed by transcript tokens based on the input audio.

Previous text tokens can be utilized as a prompt to increase the accuracy on spelling (Liao et al., 2023). For instance, when prompt is structured as 'QuirkQuid Quill', Whisper assigns high probability to the tokens of given prompt when given audio that have similar but ambiguous pronunciation, such as 'Quirk, Quid, Quill'. UKoSpeech utilizes this prompting technique to facilitate flexible switching, enhancing the versatility in various domains by improving its performance in transcribing domain-specific terminologies based on the given prompt. Specifically, we denote the prompt utilized for UKoSpeech as **domain prompt**, which comprises of 'tag' that contain domain information relevant to the transcription process. These tags are domain-specific words about the audio being transcribed. For instance, if we have information about audio being transcribed is from medical domain, we can build domain prompt as `{ domain: Prognosis, Transplant, Contusion, ... }`. Based on the information given prior to the transcription process, we explore two types of domain prompts: **domain prompt with single tag** and **domain prompt with multiple tags**.

4.1 Single Domain Tag

The single domain tag approach uses a single tag that encapsulates broad domain information pertinent to the audio. This approach is particularly effective in scenarios where minimal information about the audio content is available. The prompt format employed is `{ domain: {GENERAL_DOMAIN} }`. For instance, when transcribing audio from finance domain, the model is prompted with `{ domain: finance }`. This format is similarly applied to the medical and legal domains with the prompts `{ domain: medical }` and `{ domain: law }`, respectively.

4.2 Multiple Domain Tag

The multiple domain tag setting involves utilizing several tags that provide detailed domain-specific information about the audio content. This method is beneficial when extensive information about the audio is available prior to transcription. Given the challenge of listening to audio and manually generating tags, we utilize GPT-3.5-Turbo to automatically produce multiple domain tags based on the transcript. We formulate the prompt for GPT-3.5-Turbo as (1) instruction to generate list of domain tags from the transcript, (2) a set of examples showcasing the desired output, and (3) a transcript from our dataset. This prompt structure is detailed in Figure 2. The GPT-generated domain tags are gathered to construct domain prompt for Whisper, adhering to the format: `{ domain: {TAG_1}, {TAG_2}, ... }`.

4.3 Domain Specialized Tuning

Figure 3 illustrates our domain specialized tuning framework. Whisper is trained in multitask format, utilizing special tokens at the start of decoding stage to specify which tasks to perform. Our domain prompt is placed between the <|startofprevious|> and <|startoftranscript|> tokens, and it is trained via next-token prediction objective. During inference, the model autoregressively generates tokens with the domain prompt incorporated during the decoding stage.

4.4 Alternative Traning Approach

Despite constructing dataset that comprises of audio-text pair for training, sufficiency of domain-specific terms within the dataset is uncertain. Consequently, we investigate an alternative training approach: training Whisper using a substantial

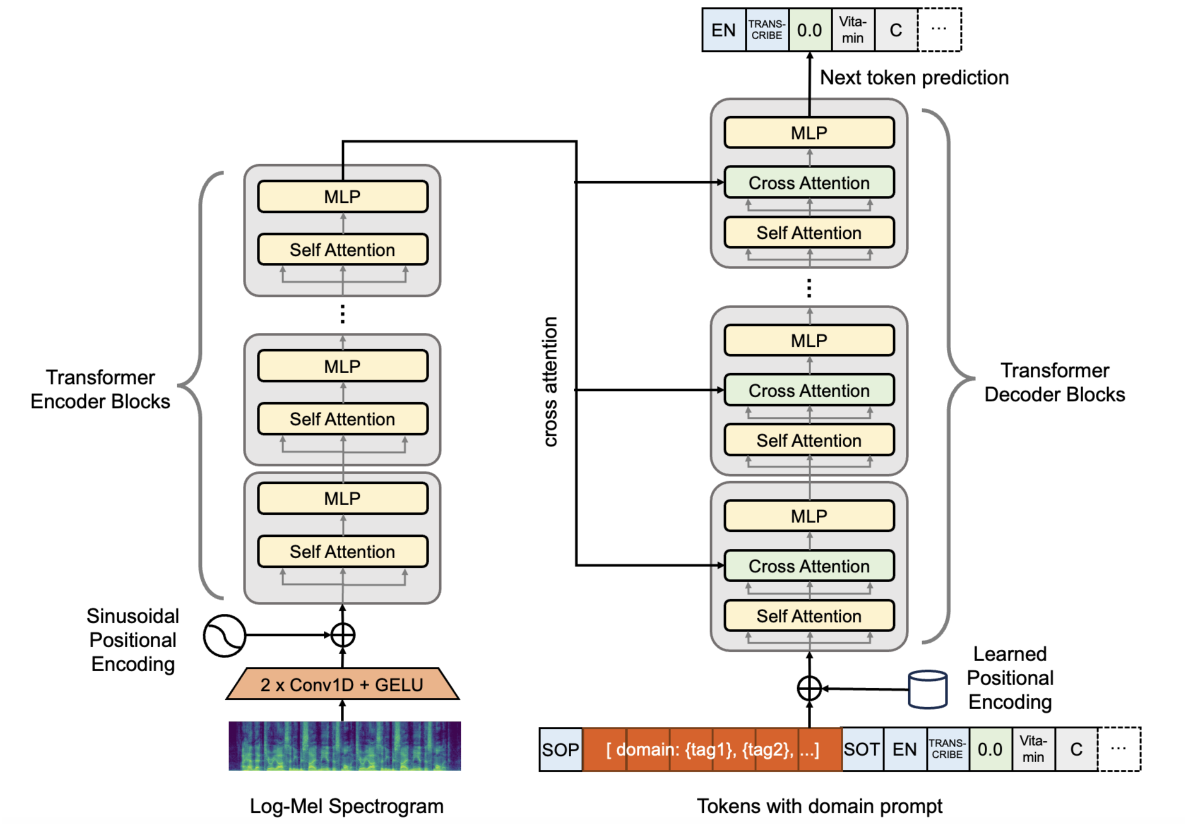

Figure 3: Overview of our training framework via paired audio-text dataset. Domain tags are prompted as initial tokens in decoding stage of Whisper. It is then trained via next-token prediction objective.

amount of text-only data, which is expected to contain a richer set of domain-specific terms. In their recent work, (Liao et al., 2023) propose a novel training method for Whisper, focusing solely on the decoder component. Specifically, they replace the cross-attention layer with a trainable bias vector, initially set to zeros, to enable the training of Whisper decoder. This bias vector operates exclusively during text-only training and is replaced by the original cross-attention layer during inference. Drawing inspiration from this methodology, we adopt a similar approach in training the Whisper decoder, substituting its cross-attention layers with bias vector. We refer to this model as **UKoSpeech-Text-Only**.

Furthermore, (Goodfellow et al.) highlighted the issue where fine-tuning a pre-trained model can lead to the model forgetting previously learned data representations, a phenomena known as catastrophic forgetting. Given that Whisper is pre-trained with paired audio-text data, catastrophic forgetting is critical to Whisper when training with text-only data. To prevent this phenomena, (Meng et al., 2022) suggests the method of training the model with paired audio-text data beforehand to make it act as a regularization. This approach motivated us to implement a "warm-up" phase for the model, where we first train it using paired audio-text data before proceeding to train with text-only data. We denote this model as **UKoSpeech-Both**.

5 Experimental Settings

5.1 Dataset

Our data curation pipeline crawled 65 YouTube channels and extracted 1,119 videos. These videos are segmented to a total number of 2,860 paired audio-text segments, amounting to 719 minutes of audio. We allocate 25% of our dataset for testing purpose and designated 20% of the remaining training data for validation during the training process. Detailed statistics of this dataset are presented in Table 4.

For text-only data, we employ two text corpus from AIHub: the "Specialized medical and legal book corpus[10]" and the "Financial legal document

[10]https://aihub.or.kr/aihubdata/data/view.do?currMenu=&topMenu=&aihubDataSe=realm&dataSetSn=

Models	WER (%)				CER (%)			
	Finance	Medical	Law	Overall	Finance	Medical	Law	Overall
Clova Speech	46.55	43.33	49.57	44.87	21.20	14.23	21.96	16.47
Google STT	35.46	33.17	42.92	35.47	15.57	12.75	20.42	14.66
Wav2vec2-xls-r-1b-korean	65.38	62.76	67.09	63.90	26.61	25.17	27.78	25.83
Whisper-large-v2	12.67	15.93	15.07	15.43	3.76	4.02	3.96	4.04
Whisper-large-v3	16.12	19.25	19.41	19.09	5.36	4.91	5.71	5.17
UKoSpeech (ours)	**11.73**	**12.61**	**14.35**	**12.74**	3.42	**3.26**	**3.84**	**3.29**
Whisper-finance-ft	11.76	-	-	-	**3.38**	-	-	-
Whisper-medical-ft	-	15.58	-	-	-	3.81	-	-
Whisper-law-ft	-	-	14.90	-	-	-	3.91	-

Table 1: Comparison of different models tested in Finance, Medical and Law domains. Bold numbers represent the least error score among each domain.

machine comprehension data[11]". We select these corpora over the dataset constructed through our data curation pipeline due to their broader and richer assortment of domain-specific terms. We segment these corpora into smaller text units to fit in the context size of Whisper decoder. Consequently, 5000 text segments for each domain are obtained.

5.2 Implementation Details

We employed Word Error Rate (WER) and Character Error Rate (CER) as the metrics to evaluate the models. To ensure an accurate comparison between the hypothesis and reference texts, we applied the KoreanNormalizer from the KoLM Python library to both texts before computing WER and CER.

Considering practical applications, we opted for the single tag domain prompt as the baseline for subsequent experiments of UKoSpeech. The impact of employing multiple tags will be discussed in Section 6.3.

We selected Whisper-large-v2, Whisper-large-v3, CLOVA Speech, Google STT and XLS-R Korean (1B), which is a XLS-R (1B)(Babu et al., 2021) fine-tuned to Zeroth Korean dataset [12] as our competitor models. Due to the lack of a publicly available leaderboard for Korean ASR models, it was not feasible to determine the top-performing model in Korea. Therefore, we chose these models for the comparison based on their widespread use. More implementation details are described in appendix A

71487

[11] https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=71610

[12] https://github.com/goodatlas/zeroth

6 Results and Discussion

6.1 Main Results

Table 1 showcases the performance of each model across various domains. The best results across each columns are highlighted as bold numbers.

Among all the models evaluated, UKoSpeech demonstrates superior performance across all tested domains. It even outperforms Whisper models that are fine-tuned on medical and law domains. This truly showcases the power of our domain prompt, eliminating the need to use separate models for each domain.

One thing to note is the unexpected performance of Whisper-large-v3. Although the large-v3 model demonstrated superior performance over large-v2 on various benchmarks, leading us to anticipate improved results on our dataset, we observed a degradation instead. We hypothesize that this unexpected outcome stems from a bias in the test data towards large-v2. In other words, transcriptions from large-v2 are served as references to filter out inappropriate texts in data curation process, thus increasing the probability of biasing toward these reference texts.

6.2 Alternative Training Approach

To discover how alternative training approach described in section 4.4 leverages the domain adaptation capability of UKoSpeech, we conducted comparative experiment between UKoSpeech-Text-Only, UKoSpeech-Both and UKoSpeech. Table 2 displays the evaluation result of models trained with different training approach. Among tested models, UKoSpeech shows the best result across all domains. On contrary, Whisper-Text-Only shows the worst performance across all domains. Our find-

Model	WER (%)				CER (%)			
	Finance	Medical	Law	Overall	Finance	Medical	Law	Overall
UKoSpeech-Text-Only	98.31	95.85	97.79	97.31	90.72	83.89	91.14	88.58
UKoSpeech-Both	16.41	20.35	20.01	19.59	5.20	5.46	5.44	5.35
UKoSpeech	**11.17**	**12.28**	**13.04**	**12.19**	**3.24**	**3.24**	**3.30**	**3.13**

Table 2: Quantitative evaluation of models trained with different methods. Best scores across each domain are bolded.

Number of Domain Tag	WER (%)				CER (%)			
	Finance	Medical	Law	Overall	Finance	Medical	Law	Overall
Whisper-large-v2								
10	23.34	16.33	25.49	17.94	14.43	6.43	15.31	8.63
1	38.12	19.61	23.91	20.55	28.01	7.14	13.13	9.34
No tag	**12.67**	**15.93**	**15.07**	**15.43**	**3.76**	**4.02**	**3.96**	**4.04**
UKoSpeech								
10	11.75	**11.56**	**13.79**	**11.90**	**3.31**	**2.98**	**3.78**	**3.07**
1	**11.73**	12.61	14.35	12.74	3.42	3.26	3.84	3.29
No tag	12.48	15.74	15.95	15.39	3.68	4.05	3.84	3.86

Table 3: Comparison of models trained with different number of domain tags.

ings suggest that the model exhibits a high degree of catastrophic forgetting, likely due to its heterogeneous modality of data used for pre-training.

For Whisper-Both, overall result is significantly better than that of Whisper-Text-Only. However, it does not reach the level of UKoSpeech. Having a closer look at the specific transcript generated by Whisper-Both, model regularly generated random characters that do not make up words when combined. We hypothesize that different modality between paired audio-text and text-only data may have caused the model to be confused within parameter space, thus generating random character.

6.3 Scalability of domain tags

This section examines (1) the potential performance enhancement of UKoSpeech with an increased number of domain tags and (2) the effectiveness of utilizing domain prompts with the base Whisper model, which is not fine-tuned with domain prompts. We compare the performance differences between UKoSpeech trained with a single domain tag versus multiple domain tags. The same domain prompt configurations are also applied to the base Whisper model. The results, as presented in Table 3, reveal that UKoSpeech trained with multiple domain tags outperforms the version with a single domain tag in most domains. In contrast, Whisper does not benefit from the addition of domain prompts; instead, it experiences a drop in performance. These findings suggest that **(1)** increasing the number of domain tags enhances the performance of UKoSpeech, and **(2)** integrating domain prompts into the base Whisper model negatively impacts its performance, emphasizing the necessity of fine-tuning with domain prompts.

7 Conclusion

We propose UKoSpeech, the Korean ASR model that specializes in diverse domains. It can flexibly change its speciality in transcribing domain-specific terms by switching its prompt. We show that UKoSpeech outperforms base Whisper in all our evaluated domains, even outperforming two most widely used ASR model in Korea, CLOVA Speech and Google STT. Our qualitative evaluation reveals that UKoSpeech accurately recognizes domain-specific terms and transcribes them correctly.

In the process of developing UKoSpeech, we designed a data curation pipeline that collects paired audio-text dataset. We believe that this pipeline (1) is adaptable for constructing datasets across additional domains, (2) has the potential to be extended into a robust framework for creating reliable paired audio-text datasets for low-resource languages.

Limitation

We face three limitations in our research: (1) Due to the lack of domain-specific dataset present in Korean, it was inevitable to conduct experiment with test dataset that have identical source as train dataset. This may have caused the overestimation in performance of UKoSpeech. (2) Although various prompt structure can be formulated such as `{This utterance is about {{GENERAL_DOMAIN}}` or `{So were just talking about {GENERAL_DOMAIN}}`, proposed in (Yang et al., 2024), we only explored the single prompt structure. Variations in performance of UKoSpeech based on different prompt structure could be further investigated. (3) Section 6.1 reveals that UKoSpeech achieves state of the art performance with WER and CER metrics. However, these metrics do not specifically guarantee the performance in transcribing domain-specific words. Therefore, using a distinct metric tailored for evaluating domain-specific words could provide a clearer demonstration.

Ethics Statement

The legality of utilizing YouTube video data for training purposes lacks precise regulations about copyright. Regarding this issue, Google has officially stated that the use of copyright-protected material is allowed under specific conditions without requiring authorization from the copyright holder[13]. They also made a statement that *The works of commentary, criticism, research, teaching, or new reporting may be considered fair use*. Nevertheless, to utterly make sure of copyright considerations, we exclusively selected videos that are licensed under "Creative Commons". This license permits contents to be reused and redistributed, allowing us to build safe and reliable dataset.

References

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. Xls-r: Self-supervised cross-lingual speech representation learning at scale.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. 2023. Whisperx: Time-accurate speech transcription of long-form audio.

Jeong-Uk Bang, Seung Yun, Seung-Hi Kim, Mu-Yeol Choi, Min-Kyu Lee, Yeo-Jeong Kim, Dong-Hyun Kim, Jun Park, Young-Jik Lee, and Sang-Hun Kim. 2020. Ksponspeech: Korean spontaneous speech corpus for automatic speech recognition. *Applied Sciences*, 10(19):6936.

Martijn Bartelds, Nay San, Bradley McDonnell, Dan Jurafsky, and Martijn Wieling. 2023. Making more of little data: Improving low-resource automatic speech recognition using data augmentation.

Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, Mingjie Jin, Sanjeev Khudanpur, Shinji Watanabe, Shuaijiang Zhao, Wei Zou, Xiangang Li, Xuchen Yao, Yongqing Wang, Yujun Wang, Zhao You, and Zhiyong Yan. 2021. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio.

Amandeep Singh Dhanjal and Williamjeet Singh. 2023. A comprehensive survey on automatic speech recognition using neural networks. *Multimedia Tools and Applications*, pages 1–46.

Saket Dingliwa, Ashish Shenoy, Sravan Bodapati, Ankur Gandhe, Ravi Teja Gadde, and Katrin Kirchhoff. 2022. Domain prompts: Towards memory and compute efficient domain adaptation of asr systems.

Ian J Goodfellow, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Raviraj Joshi and Anupam Singh. 2022. A simple baseline for domain adaptation in end to end asr systems using synthetic data. In *Proceedings of the Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pages 244–249.

Namkyu Jung, Geonmin Kim, and Joon Son Chung. 2022. Spell my name: keyword boosted speech recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6642–6646. IEEE.

Martin Kocour, Karel Veselỳ, Alexander Blatt, Juan Zuluaga-Gomez, Igor Szöke, Jan Cernockỳ, Dietrich Klakow, and Petr Motlicek. 2021. Boosting of contextual information in asr for air-traffic call-sign recognition. In *Interspeech*, pages 3301–3305.

[13]https://support.google.com/youtube/answer/9783148?hl=en

Alexander Koplenig, Sascha Wolfer, and Peter Meyer. 2023. A large quantitative analysis of written language challenges the idea that all languages are equally complex. *Scientific Reports*, 13(1):15351.

Egor Lakomkin, Sven Magg, Cornelius Weber, and Stefan Wermter. 2018. Kt-speech-crawler: Automatic dataset construction for speech recognition from youtube videos. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 90–95.

Jinyu Li et al. Recent advances in end-to-end automatic speech recognition. *APSIPA Transactions on Signal and Information Processing*, 11(1).

Feng-Ting Liao, Yung-Chieh Chan, Yi-Chang Chen, Chan-Jan Hsu, and Da-shan Shiu. 2023. Zero-shot domain-sensitive speech recognition with prompt-conditioning fine-tuning. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8. IEEE.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Zhong Meng, Yashesh Gaur, Naoyuki Kanda, Jinyu Li, Xie Chen, Yu Wu, and Yifan Gong. 2022. Internal language model adaptation with text-only data for end-to-end speech recognition.

Gayani Nanayakkara, Nirmalie Wiratunga, David Corsar, Kyle Martin, and Anjana Wijekoon. 2022. Clinical dialogue transcription error correction using seq2seq models.

Chanjun Park, Jaehyung Seo, Seolhwa Lee, Chanhee Lee, Hyeonseok Moon, Sugyeong Eo, and Heui-Seok Lim. 2021. 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.

Chanjun Park, Jaehyung Seo, Seolhwa Lee, Junyoung Son, Hyeonseok Moon, Sugyeong Eo, Chanhee Lee, and Heui-Seok Lim. 2024. Hyper-bts dataset: Scalability and enhanced analysis of back transcription (bts) for asr post-processing. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 67–78.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society.

Vineel Pratap, Qiantong Xu, Jacob Kahn, Gilad Avidov, Tatiana Likhomanenko, Awni Hannun, Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert. 2020. Scaling up online speech recognition using convnets.

Ofir Press, Noah Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR.

Shinnosuke Takamichi, Ludwig Kürzinger, Takaaki Saeki, Sayaka Shiota, and Shinji Watanabe. 2021. Jtubespeech: corpus of japanese speech collected from youtube for speech recognition and speaker verification.

Juan Camilo Vásquez-Correa, Haritz Arzelus, Juan M Martin-Doñas, Joaquin Arellano, Ander Gonzalez-Docasal, and Aitor Álvarez. 2023. When whisper meets tts: Domain adaptation using only synthetic speech data. In *International Conference on Text, Speech, and Dialogue*, pages 226–238. Springer.

Chao-Han Huck Yang, Bo Li, Yu Zhang, Nanxin Chen, Rohit Prabhavalkar, Tara N Sainath, and Trevor Strohman. 2023. From english to more languages: Parameter-efficient model reprogramming for cross-lingual speech recognition. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Chih-Kai Yang, Kuan-Po Huang, and Hung yi Lee. 2024. Do prompts really prompt? exploring the prompt understanding capability of whisper.

Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, Zhong Meng, Ke Hu, Andrew Rosenberg, Rohit Prabhavalkar, Daniel S. Park, Parisa Haghani, Jason Riesa, Ginger Perng, Hagen Soltau, Trevor Strohman, Bhuvana Ramabhadran, Tara Sainath, Pedro Moreno, Chung-Cheng Chiu, Johan Schalkwyk, Françoise Beaufays, and Yonghui Wu. 2023. Google usm: Scaling automatic speech recognition beyond 100 languages.

Ding Zhao, Tara N Sainath, David Rybach, Pat Rondon, Deepti Bhatia, Bo Li, and Ruoming Pang. 2019. Shallow-fusion end-to-end contextual biasing. In *Interspeech*, pages 1418–1422.

A Implementation Details

We established Whisper-large-v2 as the baseline for UKoSpeech. For extraction of audio features, we followed identical sampling rate and number of mel features as our baseline model, which is 16kHz and 80, respectively. During the training process, we employed LoRA (Hu et al., 2021) to accommodate the limited resources available for training. Batch size of 4 and AdamW (Loshchilov and Hutter, 2019) optimizer with an initial learning rate of 5e-5 are utilizeds. The training spanned two epochs.

Furthermore, to explore the efficacy of domain prompts, we compared UKoSpeech with (1) Whisper-large-v2 fine-tuned to our dataset, denoted as **Whisper-large-v2-ft** (2) XLS-R-Korean fine-tuned to our dataset, denoted as **XLS-R-Korean-ft** and (3) Whisper-large-v2 fine-tuned to different domains from our dataset, denoted as **Whisper-[chosen domain]-ft**. Since Whisper-[chosen domain]-ft are tailored to specific domains, we evaluate their performance only within the domains they were trained on. Data statistics are shown in Table 4.

Domain	**Number of data**		**Length of audio (min)**	
	Train	**Test**	**Train**	**Test**
Finance	197	58	37	11
Medical	1491	370	401	101
Law	600	144	135	31
Total	**2288**	**572**	**576**	**143**

Table 4: Statistics of our dataset. Length of audio data is described in minute. 20% of train data are utilized as validation data.

We conduct a quantitative evaluation of each model and a detailed qualitative analysis of how each model transcribes identical domain-specific terminologies, which is detailed in Appendix B.

B Qualitative Analysis

Table 5 illustrates how each model transcribes domain-specific terminologies. UKoSpeech is the only model that consistently transcribes all terms correctly. While CLOVA Speech and Google STT generally transcribe domain-specific terms accurately, we observed that they occasionally cease sentence generation or omit certain segments of the audio. This behavior may have contributed to their underwhelming performance in previous quantitative experiment.

Models	Finance	Medical	Law
Reference	스케일업 (Scale-Up) [*seukeil-eob*]	경동맥 내막절제술 (Carotid endarterectomy) [*gyeongdongmaeg naemagjeoljesul*]	임대차 보증금 (Rental deposit) [*imdaecha bojeung-geum*]
Naver Clova	스케일업	경동맥 내막 절제술	보증금 [*bojeung-geum*]
Google STT	스케일 업	경동맥 내막 절제술	차 보증금 [*cha bojeung-geum*]
Wav2vec2-xls-r-1b-korean	스케 [*seuke*]	경동 백 네막 절체소 [*gyeongdong baeg nemag jeolcheso*]	자보증금 [*jabojeung-geum*]
Whisper-large-v2	스케이럽 [*seukeileob*]	경동맥 뇌막 절제술 [*gyeongdongmaeg noemag jeoljesul*]	2대차 보증금 [*idaecha bojeung-geum*]
Whisper-large-v3	스케이럽 [*seukeileob*]	경동맥 뇌막 절제술 [*gyeongdongmaeg noemag jeoljesul*]	임대차보증금
UKoSpeech (ours)	스케일업	경동맥 내막절제술	임대차 보증금
Whisper-large-v2-ft	-	-	-
Whisper-finance-ft	스케일업	-	-
Whisper-medical-ft	-	경동맥 뇌막 절제술 [*gyeongdongmaeg noemag jeoljesul*]	-
Whisper-law-ft	-	-	2대차 보증금 [*idaecha bojeung-geum*]

Table 5: Qualitative evaluation on domain-specific terminologies. Word inside bracket indicate the meaning of each words. Words transcribed correctly by each model are highlighted in green, while those incorrectly transcribed are shown in red. We mark each pronunciation with *Italic* font.