

Method and System for Automatic Speech Recognition Correction

Field of the Invention

The present application relates to methods and systems for Automatic Speech Recognition (ASR) correction, and in particular constructing training datasets and using the datasets to train a plurality of error correction models for correcting speech transcriptions created through ASR.

Background

In the burgeoning field of automatic speech recognition (“ASR”) applications, it is often necessary to correct speech recognition errors due to factors like environmental noise and variations in user speech or pronunciation. ASR Error Correction (“EC”) refers to a variety of techniques which may be applied by researchers and operators of machine learning systems to lower Word Error Rates (WER) and the improve the overall accuracy of outputted text transcriptions from ASR systems. EC processing typically relies on large amounts of training data including parallel data such as pairs of correct sentences (reference transcripts) and incorrect sentences (ASR transcripts) corresponding to the same speech signal. This parallel data is used to determine and correct for errors that are found in the ASR transcripts.

To minimize speech recognition errors, some known approaches use large-scale text only data to correct ASR outputs. However, such approaches are ill-suited due to potential variations or irregular patterns of errors corresponding to the context, pronunciation, and language of using text only reference information. Due to such errors, another approach is to utilize transcripts corresponding to audio samples for EC. However, in practice, certain errors cannot be sufficiently corrected with audio and text information. For example, the spoken sentence or phrase “if you plan a little bit” and “if you plant a little bit” are both grammatically correct and also sound quite similar to one another. Consequently, for such sentences, using audio and text information would not be sufficient for correcting the output of the ASR system and lowering its WER. While there are known ASR systems or lip-reading detection models that use visual information, the performance gain of such systems is limited due to the quantity and scale of the datasets used for pretrained speech recognition.

Thus, it is desirable to provide an ASR error correction model to address the disadvantage of the existing art and/or to provide the public with a useful choice.

Summary of the Invention

The present invention aims to provide new and useful methods and systems for Automatic Speech Recognition (ASR) correction, and in particular constructing training datasets including a combination of visual, audio and speech information and using the constructed
5 datasets to train machine translation models for correcting speech transcriptions created through ASR.

In broad terms, the present invention proposes a speech recognition error correction system suitable to receive candidate text transcriptions associated with captured speech information from an automatic speech recognition (ASR) system and use a machine learning model to
10 output corrected candidate text transcriptions based on a filtered dataset and received candidate text transcriptions. One way of implementing this is extracting image data and their corresponding reference transcriptions associated with a plurality of image frames of a dataset of videos and filtering the extracted image data based on a similarity calculation between the image data and their reference transcriptions to obtain a filtered dataset
15 including one or more filtered reference transcriptions and their corresponding image data. The filtered reference transcriptions may correspond to a set of extracted textual transcripts which are determined to require correction based on their associated image data. As it can be appreciated from the described embodiment, the ASR error correction method or model may leverage large-scale and varied modalities of training data including a combination of
20 visual, speech, and textual information, for achieving improved WER and ASR text correction.

In one aspect, the extracted image data corresponds to a generated caption data of a video frame and the reference transcriptions correspond to grammatically accurate textual transcripts of a video frame. The textual transcripts may relate to recorded audio data from
25 the extracted video frame. In implementations, the candidate text transcriptions may be obtained from a baseline transformers model pre-trained by using synthetic datasets generated by a computer simulation.

A second aspect of the invention relates to a method and system of training a plurality of machine learning models using the filtered dataset and candidate text transcriptions. The
30 machine learning models may be transformer-based models.

The extracted image data may be inputted into a visual language model trained for image captioning to generate caption data as output corresponding to the image data. Image data associated with the filtered set of reference transcriptions may be optionally input into a vision transformer for obtaining a vector representation of the image data.

In implementations, the machine learning model comprises a prompt-based neural training model or a fusion-based training model. A weak checkpoint trained on a transcribed dataset may be used to obtain synthetic datasets corresponding to the candidate text transcriptions.

Optionally, the synthetic datasets may be obtained from baseline machine-learning models and the baseline machine-learning models may be further refined using the filtered dataset.

The similarity calculation may be determined using a sentence transformer model. A byte-pair encoding (BPE) algorithm may be applied during training of the sentence transformer model.

The caption data and the candidate text transcriptions may be grouped as concatenated pairs and the caption data and the candidate text transcriptions can also be used for training the machine learning model.

Each of the concatenated pairs may contain separator tokens for separating the caption data from the candidate text description.

The candidate text transcriptions may be input into a text transformer to obtain vector representations of the candidate text transcriptions.

In some implementations, vector representations of the image data and candidate text transcriptions may be fused by a vector concatenation process to obtain a concatenation vector. The concatenation vector of the vector representations can be passed through a gate to control amount of visual information used. The concatenation vector can be further added to the vector representation of the candidate text transcription to generate a multimodal fusion representation, i.e. an output fusion vector, wherein the multimodal fusion representation is decoded to generate the corrected candidate text transcription.

Corrected candidate text transcriptions may be used to update the candidate text transcriptions in a post-processing step of a fusion-based training model based on evaluating the similarity of the corrected candidate text transcriptions to the extracted image data as compared to the similarity of the extracted image data to the candidate text transcriptions. In cases where the similarity of the determined corrected candidate text transcriptions to the image data is lower than the similarity of the original candidate text transcriptions to that of the image data, the original candidate text transcriptions would not be corrected in the fusion-based training model. The corrected candidate text transcriptions may be determined from the transformer output of the prompt-based training model or the fusion-based training model.

The invention may be expressed as a method, or alternatively as a system comprising one or more computers and one or more storage devices storing instructions that when executed by the one or more computers cause the one or more computers to perform the method. It may also be expressed as a computer program product, such as downloadable program instructions (software) or one or more non-transitory computer storage media storing instructions. When executed by one or more computers, the program instructions cause the one or more computers to perform the method.

Brief Description of the Drawings:

Embodiments of the invention will now be explained for the sake of example only, with reference to the following figures in which:

Figure 1 is flow diagram of a construction of a training dataset, according to an embodiment the present disclosure.

Figure 2 is a flow diagram of using a prompt-based training system for processing the dataset of Figure 1 for ASR error correction, according to an embodiment of the present disclosure.

Figure 3 is a flow diagram of a fusion-based training system for processing the dataset of Figure 1, according to an embodiment the present disclosure.

Figure 4 is a block diagram of a prompt-based training system, according to an embodiment the present disclosure.

Figure 5 is a block diagram of a fusion-based transformer system, according to an embodiment the present disclosure.

Figure 6 is a block diagram illustrating an example computer system which can be used to perform the methods of Figures 1-3, and to implement the systems of Figures 4 and 5.

Detailed Description of Embodiments

Figure 1 is a flow diagram 100 of constructing a training dataset for training machine learning models for accurate and robust Automatic Speech Recognition (ASR) correction, in accordance with some embodiments of the present disclosure.

Although process or method steps may be described in a sequential order, such processes, methods and techniques may be configured to work in alternate orders. In other words, any sequence or order of steps that may be described does not necessarily indicate a

requirement that the steps to be performed in that order. The steps of processes described herein may be performed in any order practical. Further, some steps may be performed simultaneously.

Referring to the steps of the method 100, at step 105, the processor 622 is configured to
5 extract datasets of image or video data and candidate ASR transcripts from one or more large-scale dataset repositories. The ASR transcripts may include textual transcriptions of an audio track related to each keyframe of an extracted video or image. The textual transcriptions may have been obtained through known open-source speech recognition programs by translating the audio to text. An example of a large-scale dataset repository
10 with image and context data is a dataset which includes 300 hours or more of videos and annotated transcripts in English for each video in the dataset. Image data for the datasets may be obtained by capturing the frame of each video sample at exactly the middle of the start and end timestamps for each video sample.

In step 110, to obtain a high-quality annotated dataset of image and reference data, the
15 sample datasets are filtered based on a determined similarity between caption data associated with the image data and extracted reference transcriptions of speech (such as grammatically correct sentences). Caption data may include generated text transcriptions corresponding to inputted images or video data sources. For example, the extracted reference transcriptions may correspond to extracted grammatically accurate textual
20 transcriptions associated with the audio track of each video frame in the dataset. The caption data may correspond to extracted image data corresponding to each frame of video or image of the dataset. Prior to determining the similarity of the datasets, the datasets may be pre-processed such that the caption data and the textual transcriptions are both provided in the same language, such as English. The similarity calculation, in step 110, is preferably
25 performed using a sentence-transformer based model. An example python code of the determined similarity score between a target sentence (i.e. grammatically correct sentence) and a transcription of a caption or subtitle of an image using the sentence-transformer is provided below:

```
from sentence_transformers import SentenceTransformer, util  
30 model = SentenceTransformer('sentence-transformers/all-MiniLM-L6-v2')  
embedding_1= model.encode(caption, convert_to_tensor=True)  
embedding_2 = model.encode(target, convert_to_tensor=True)  
score = util.pytorch_cos_sim(embedding_1, embedding_2).item()
```

For example, to obtained a high-quality annotated dataset, the repository datasets may be
35 filtered to include only those datasets where the similarity score between the target sentence

and the caption transcription is greater than or equal to 0.2. Accordingly, in one implementation, a filtering of 220,000 samples of a repository dataset and approximately another 2.5 million samples of videos based a similarity score of greater than or equal to 0.2 generates a robust dataset of approximately 58,000 sample datasets and 127,000 sample datasets.

Along with determining the filtered dataset based on a similarity calculation, the training set can optionally include a number of synthetic baseline datasets for deriving candidate ASR transcripts, as provided in step 115. Synthetic datasets may be manufactured artificially using computer algorithms rather than datasets that are extracted from images or objects from the real world. As an example, the baseline or candidate datasets may be obtained from online sources such as TED-LIUM3 (videos selected from TED conferences), DATA2 (datasets build for end-to-end name entity recognition tasks), or LibriSpeech (datasets derived from audiobooks that are part of the LibriVox project). The baseline and synthetic ASR transcripts may correspond to grammatically incorrect sentences or phrases that may require additional context for correction. The synthetic ASR transcripts may be derived by using a weak checkpoint trained on the dataset. Using this extraction method, up to 5 million synthetic datasets of candidate ASR transcripts can be obtained. These generated candidate datasets would most likely be grammatically inaccurate due to the implementation of one or more weak pre-trained checkpoints for the derivation of the datasets. In step 120, the constructed resultant dataset including the filtered dataset of step 110 and the candidate datasets of transcripts of step 115 is sent to one or more multi-modal transformers for further processing and ASR error correction. A multi-modal transformer, such as the prompt-based training system of Figure 4 or the fusion-based transformer system of Figure 5, is typically configured to process input image related data such as caption data, and candidate ASR transcriptions for generating as output corrected ASR transcriptions.

Figure 2 is a flow diagram 200 for a prompt-based training model, an example of the multi-modal transformer for processing the constructed resultant dataset of Figure 1. Figure 4 is a block diagram 400 of the prompt-based training model, in accordance with an embodiment of the present disclosure.

Referring to the steps of the method 200, at step 205, processor 622 is configured to retrieve the filtered dataset and dataset of synthetic transcripts of step 120 in order to further train the prompt-based training system 400. In this example, an image caption is generated to provide additional context to the sentence or phrase for input into transformer 415 for performing ASR error correction. The generated image caption data may be text transcriptions corresponding to image data of an image frame.

In step 210, the image data corresponding to the filtered dataset is inputted into a Visual Language Modelling (VLM) system 405 for generating image captions such as descriptive sentences or phrases or summaries associated with the image data.

In step 215, the generated image caption data is input into tokenizer 410 along with the candidate ASR transcripts. The tokenizer 410 is used to transform the text or words of the caption data and the candidate ASR transcripts into a series of tokens for further processing (i.e. converted to token-ids) by the transformer encoder-decoder system 415. According to one embodiment, the tokenizer may use Byte-Pair-Encoding (BPE) to convert each word or sub-word into input IDs from a lookup table. For example, if the determined caption data is “the *tortoise* is being care for” (corresponding to an image of a tortoise being held by an owner) and the paired candidate ASR transcript is “that’s really how we choose our *tourist* today” then a modified source sentence (i.e. paired sequence) inputted into the tokenizer is “the *tortoise* is being cared for [SEP] that’s really how we choose our *tourist* today” where the [SEP] corresponds to a special token used to separate the image caption description from the candidate ASR transcript. The use of the [SEP] token to separate the image caption from the candidate ASR transcript allows transformer system 415 to distinguish the image caption as a prompt or additional available context in the prompt-based system for correcting the term “tourist” to tortoise” in the above example.

In step 220, tokenized pairs of the image caption and the ASR transcript are input into the transformer system 415 for training of the transformer 415 and generating of a corrected candidate ASR transcript. Some transformers are known for their effectiveness in natural language processing. For example, a preferred transformer encoder-decoder model 415 may be a standard BART (Bidirectional and Auto-Regressive Transforms) model.

Following supervised training of the transformer model, the corrected candidate transcript (for example: that’s really how we choose our *tortoise* for today”) is output by the transformer 415 and displayed to a user.

Figure 3 is a flow diagram 300 for a fusion-based training model, an example of the multi-modal transformer system for processing the constructed resultant dataset of Figure 1.

Figure 5 is a block diagram 500 of the fusion-based training model, in accordance with an embodiment of the present disclosure.

Referring to the steps of the method 300, at step 305, processor 622 is configured to retrieve the filtered dataset and dataset of candidate ASR transcripts of step 120 in order to train the fusion-based training system 500. In this example multi-modal training model, an integration of image and text related modalities can be achieved for error correction of ASR.

In step 310, the text data from the filtered dataset, i.e. the filtered set of reference transcripts of speech from the filtered dataset, and the ASR candidate transcripts may be first pre-processed using a Byte-Pair-Encoding (BPE) compression algorithm. BPE is a type of data compression algorithm where the most common pair of consecutive bytes of data may be replaced with a byte that does not occur with that data. In preferred embodiments, the text data may be pre-processed using a GPT-2 transformer based BPE.

In step 315, the text data pre-processed using BPE and the image data from the filtered dataset are further processed to obtain vector representations of each type of data. The vision encoder 505 is used to transform the image data into a vector representation or “image features” of the image data. In some embodiments, the image features may be obtained using a Vision Transformer (ViT) model. The text encoder 510 may follow a general transformer encoder-based paradigm for natural language processing (NLP) such as BERT. The text encoder 510 is used to transform the pre-processed text data into a vector representation or “text feature” of the text data.

In step 320, the outputted image and text features, i.e. the vector representations of the image and text data, are fused by a vector concatenation process to generate a fusion vector using a gated-fusion network 515 for fusing the image and text representations. For example the two representations may be fused by a vector concatenation according to: $H^{fused} = [H^s ; H^I]$ where H^I is a projected form of the Vision Transformer model ViT(I) to that of the length of the text representation H^s (obtained from a standard transformer-based encoder). Using the gated-fusion network 515, the fused representation H^{fused} may then be passed through a tanh gate to control the amount of visual information used as $\Lambda = \tanh(f([H^s ; H^{fused}]))$. The obtained gated-fused information Λ is then added to the outputted text representations to obtain a multi-modal fusion representation as: $H^{out} = H^s + \Lambda H^{fused}$.

The outputted fusion vector H^{out} is then inputted into a decoder 520 of the fusion-based system 500 for decoding the encoded visual and textual information for obtaining the error corrected candidate text.

The candidate text transcription may not be corrected if the determined similarity score between the candidate text transcript and the image caption data is greater than the similarity score of the corrected candidate transcript outputted from decoder 520 and the image caption data. The similarity scores may be calculated using the ViT model.

We now turn to the description of various experiments used to evaluate embodiments of the disclosure. Some of these illustrate embodiments of the disclosure other than those discussed above.

The experiments were conducted using two different ASR models, Google ASR API and Huggingface wav2vec to verify the generality of the methods. The experiments use a filtered annotated dataset with images of a generally smaller size (185,000 samples) to correct grammatical errors which is applied sequentially after first obtaining the results from the baseline transformers model (as shown as “Transformers + Gated fusion” in Table 1 below).

The baseline transformers model used in the experiments was standard BART-base model trained using Fairseq. This model was first pre-trained using 5 million synthetic data obtained from TED-LIUM3, DATA2, and LibriSpeech and fine-tuned with the filtered annotated dataset of 185,000 samples. GPT-2 based BPE was applied in the pre-processing stage. For the prompt-based method, caption data was used as prompts in the source sentence in all three train, valid and test sets. Changes made by the gated-fusion method to the output from the baseline method were filtered by including only those changes where the similarity probability between the image and the changed text is higher than that with the original text. The original text is the output from the baseline method as shown in Table 1 and this filtering is referred to as Transformers + Gated fusion (Filter) in Table 1. The similarity probability of the experiments is measured by the ViT model.

Similar filtering and sequential correction are also applied after the caption prompt-based and gated-fusion based method. These methods are referred to as Prompt-based + Gated Fusion and Prompt-based + Gated Fusion (Filter) respectively in Table 1. For the Gated Fusion method, all the testing parameters were kept constant, besides changing of the learning rate to 0.001, the max updates to 800,000, and evaluating the average of the last 10 checkpoints which yield more reliable results.

ID	Models and Variants	Google ASR API		Huggingface wav2vec	
		WER	SER	WER	SER
1	Original	36.80	100	31.94	97.79
2	Transformer	34.14	99.42	22.36	91.45
3	Prompt-based	33.5	98.94	21.13	90.59
4	Gated fusion	36.58	99.90	25.44	95.39
5	Transformer + Gated fusion	34.8	99.42	22.92	92.8
6	Transformer + Gated fusion (Filter)	34.54	99.42	22.66	92.51
7	Prompt-based + Gated Fusion	34.21	98.94	21.80	92.12
8	Prompt-based + Gated Fusion (Filter)	33.9	99.04	21.49	91.74
9	Random Image captions(Caption Prompt)	34.2	99.14	22.11	91.16
10	Original(Random test set)	34.59	99.56	30.70	97.6
11	Transformer(Random test set)	32.49	98.68	21.48	90.4
12	Prompt-based (Random test set)	30.79	97.8	20.6	89.10

Table 1: Measurement of error correction performance

It can be seen that the best results are obtained from the embodiment of the prompt-based method, thus verifying that visual information is helpful for ASR Error Correction. To study the relevance of image information in ASR correction, further experiments were conducted by assigning a random image to the parallel data and seeing if it can help in the ASR correction. As we can see from Table 1, including random image's caption as prompt to the transformer model, leads to a decrease in performance of WER from the prompt-based method, and is on par with the baseline transformer model. This shows that the right image of the video captured during the speech is essential to improve EC performance.

Because it's easier to use large amounts of synthetic data (i.e. parallel data with images) for text-only methods, the prompt-based method provides robust performance. Additional experiments were performed with an increase in synthetic data as provided in Table 2:

ID	Method	2M Synthetic		5M Synthetic	
		WER	SER	WER	SER
1	Prompt-based	33.94	99.33	33.5	98.94

Table 2: EC performance with different size of synthetic data

It was found also that the prompt-based method shows a better representation of the caption data as context data (and thus, achieves better ASR error correction) when the amount of synthetic data to be learnt increases for the model.

Furthermore, it was found that the caption prompt method has potential to also improve performance of ASR text correction for instructional videos (i.e. not only for a test set having examples where images are needed for correction). A test set of 1000 random sentences from the dataset was sampled and the results of table 1 show the performance gap of the baseline and the caption prompt methods increases by 2x for google ASR API and decreases only slightly for huggingface wav2vec.

Many other possible variations of the method explained above are also possible within the scope of the invention. For example, in some variations, adding additional information of audio data resulting in improved captioning data and potential improvements to performance.

As used in this application, the terms "component," "module," "engine," "system," "apparatus," "interface," or the like are generally intended to refer to a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution. For example, a component may be, but is not limited to being, a process running on a processor, a processor, an object, an executable, a thread of execution, a program,

and/or a computer. By way of illustration, both an application running on a controller and the controller can be a component. One or more components may reside within a process and/or thread of execution and a component may be localized on one computer and/or distributed between two or more computers.

- 5 Furthermore, the claimed subject matter may be implemented as a method, apparatus, or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof to control a computer to implement the disclosed subject matter. For instance, the claimed subject matter may be implemented as a computer-readable medium embedded with a computer executable
- 10 program, which encompasses a computer program accessible from any computer-readable storage device or storage media. For example, computer readable media can include but are not limited to magnetic storage devices (e.g., hard disk, floppy disk, magnetic strips . . .), optical disks (e.g., compact disk (CD), digital versatile disk (DVD) . . .), smart cards, and flash memory devices (e.g., card, stick, key drive . . .).
- 15 Figure 6 is a block diagram showing the technical architecture 600 of a server which can perform some or all of the methods of Figures 1-3. The technical architecture includes a processor 622 (which may be referred to as a central processor unit or CPU) that is in communication with memory devices including secondary storage 624 (such as disk drives), read only memory (ROM) 626, random access memory (RAM) 628. The processor 622 may
- 20 be implemented as one or more CPU chips. The technical architecture may further comprise input/output (I/O) devices 630, and network connectivity devices 632.

The secondary storage 624 is typically comprised of one or more disk drives or tape drives and is used for non-volatile storage of data and as an over-flow data storage device if RAM 628 is not large enough to hold all working data. Secondary storage 624 may be used to

25 store programs which are loaded into RAM 528 when such programs are selected for execution.

In this embodiment, the secondary storage 624 has an order processing component 624a comprising non-transitory instructions operative by the processor 622 to perform various operations of the method of the present disclosure. The ROM 626 is used to store

30 instructions and perhaps data which are read during program execution. The secondary storage 624, the RAM 628, and/or the ROM 626 may be referred to in some contexts as computer readable storage media and/or non-transitory computer readable media.

I/O devices 630 may include printers, video monitors, liquid crystal displays (LCDs), plasma displays, touch screen displays, keyboards, keypads, switches, dials, mice, track balls, voice

35 recognizers, card readers, paper tape readers, or other well-known input devices.

The processor 622 executes instructions, codes, computer programs, scripts which it accesses from hard disk, floppy disk, optical disk (these various disk-based systems may all be considered secondary storage 624), flash drive, ROM 626, RAM 628, or the network connectivity devices 632. While only one processor 622 is shown, multiple processors may be present. Thus, while instructions may be discussed as executed by a processor, the instructions may be executed simultaneously, serially, or otherwise executed by one or multiple processors.

Although the technical architecture is described with reference to a computer, it should be appreciated that the technical architecture may be formed by two or more computers in communication with each other that collaborate to perform a task. For example, but not by way of limitation, an application may be partitioned in such a way as to permit concurrent and/or parallel processing of the instructions of the application. Alternatively, the data processed by the application may be partitioned in such a way as to permit concurrent and/or parallel processing of different portions of a data set by the two or more computers.

In an embodiment, virtualization software may be employed by the technical architecture 600 to provide the functionality of a number of servers that is not directly bound to the number of computers in the technical architecture 600. In an embodiment, the functionality disclosed above may be provided by executing the application and/or applications in a cloud computing environment. Cloud computing may comprise providing computing services via a network connection using dynamically scalable computing resources. A cloud computing environment may be established by an enterprise and/or may be hired on an as-needed basis from a third-party provider.

By programming and/or loading executable instructions onto the technical architecture, at least one of the CPU 622, the RAM 628, and the ROM 626 are changed, transforming the technical architecture in part into a specific purpose machine or apparatus having the novel functionality taught by the present disclosure. It is fundamental to the electrical engineering and software engineering arts that functionality that can be implemented by loading executable software into a computer can be converted to a hardware implementation by well-known design rules.

Whilst the foregoing description has described exemplary embodiments, it will be understood by those skilled in the art that many variations of the embodiment can be made within the scope and spirit of the present invention.

Claims:

1. A speech recognition error correction method comprising:
receiving candidate text transcriptions associated with captured speech information
5 from an automatic speech recognition (ASR) system;
extracting image data and reference transcriptions associated with a plurality of
image frames of a dataset of videos;
filtering the extracted image data based on a similarity calculation between the image
data and the reference transcriptions to obtain a filtered dataset including a filtered set of
10 reference transcriptions and their corresponding image data; and
using a machine learning model to output corrected candidate text transcriptions
based on the filtered dataset and the received candidate text transcriptions.
2. The method according to claim 1, wherein the image data corresponds to caption
data associated with an image input of a video frame and the extracted reference
15 transcriptions corresponds to recorded audio data of one or more video frames.
3. The method according to any of claims 1-2, wherein the extracted reference
transcriptions includes one or more grammatically accurate textual transcripts related to the
audio of one or more image frames.
4. The method according to any of claims 1-3, wherein the candidate text transcriptions
20 are obtained from a baseline transformers model pre-trained by using one or more synthetic
datasets generated by at least one computer simulation.
5. The method according to any of claims 1-4, wherein the filtered dataset and the
candidate text transcriptions correspond to input data inputted to the machine learning model
for training the machine learning model.
- 25 6. The method according to any of claims 1-5, wherein the extracted image data is
inputted into a visual language model trained for image captioning to generate caption data
as output corresponding to the image data.
7. The method according to any of claims 1-6, wherein image data associated with the
filtered dataset is input into a vision transformer for obtaining a vector representation of the
30 image data.
8. The method according to any of claims 1-7, wherein the machine learning model
comprises at least one of a prompt-based training model or a fusion-based training model.

9. The method of according to any of claims 1-8, wherein a weak checkpoint trained on a transcribed dataset is used to obtain one or more synthetic datasets corresponding to the candidate text transcriptions.
10. The method according to claim 9, wherein the one or more synthetic datasets are
5 obtained from one or more baseline machine-learning models.
11. The method according to claim 10, wherein the one or more baseline machine-learning models are further refined using the filtered dataset.
12. The method according to any of claims 1-11, wherein the similarity calculation is determined using a sentence transformer model.
- 10 13. The method according to claim 12, wherein a byte-pair encoding (BPE) algorithm is applied during training of the sentence transformer model.
14. The method according to claim 6, wherein each of the caption data and the candidate text transcriptions are grouped as one or more concatenated pairs.
15. The method according to any of claim 6 or 14, wherein the caption data and
15 candidate text descriptions are used for training the machine learning model.
16. The method of according to any of claims 1-15, wherein the machine learning model is a transformer-based model.
17. The method according to claim 14, wherein each of the concatenated pairs contain one or more separator tokens for separating the caption data from the candidate text
20 description.
18. The method according to claim 17, wherein the candidate text transcriptions are input into a text transformer to obtain vector representations of the candidate text transcriptions.
19. The method according to claim 18, wherein vector representations of the image data and candidate text transcriptions are fused by a vector concatenation process to obtain a
25 concatenation vector.
20. The method according to claim 19, wherein the concatenation vector is further processed to generate the corrected candidate text transcriptions.
21. The method according to claim 1, wherein the corrected candidate text transcriptions are used to update the candidate text transcriptions in a post-processing step of a fusion-based training model based on evaluating the similarity of the corrected candidate text
30 transcriptions to the extracted image data as compared to the similarity of the extracted image data to the candidate text transcriptions, wherein the corrected candidate text

transcriptions are determined based on either the prompt-based training model or the fusion-based training model.

22. The method according to claim 19, wherein the concatenation vector is passed through a gate to control amount of visual information used.

5 23. The method according to claim 19, wherein the concatenation vector is added to the vector representation of the candidate text transcription to generate a multimodal fusion representation corresponding to an output fusion vector, wherein the multimodal fusion representation is decoded to generate the corrected candidate text transcription.

10 24. A system comprising one or more computers and one or more storage devices storing instructions that when executed by the one or more computers cause the one or more computers to perform the method of any one of claims 1-23.

25. One or more non-transitory computer storage media storing instructions that when executed by one or more computers cause the one or more computers to perform the operations of any of claims 1-24.

15

20

25