

DEVICE AND METHOD FOR TRAINING A MACHINE LEARNING MODEL TO EXTRACT TEXT ITEMS FROM TEXT PHOTOS

TECHNICAL FIELD

[0001] Various aspects of this disclosure relate to devices and methods for training a machine learning model to extract text items from text photos.

BACKGROUND

[0002] Many texts are more easily available in image (i.e. photo form) than in editable form, for example menus of restaurants. However, for further processing of the text, it needs to be converted to another (editable form) which may cause a high amount of manual labour, e.g. to bring menus of multiple restaurant into an editable form such that they can be published on the site of a food delivery service. A machine learning model may be applied for such a task but it needs to be properly trained to provide accurate results.

[0003] Accordingly, effective approaches for training a machine learning model to extract text items from text photos are desirable.

SUMMARY

[0004] Various embodiments concern a method for training a machine learning model to extract text items from text photos, comprising extracting a text item from a text photo using the machine learning model, determining a correspondence between the extracted text item and an image including a region of the text photo showing text from which the machine learning model has extracted the text item by means of a generative artificial intelligence model and adjusting the machine learning model depending on the determined correspondence.

[0005] According to one embodiment, adjusting the machine learning model depending on the determined correspondence comprises adjusting a processing of the text photo by the machine learning model depending on the determined correspondence.

[0006] According to one embodiment, adjusting the machine learning model depending on the determined correspondence comprises adjusting the machine learning model to improve the correspondence if the correspondence is below a predetermined threshold.

[0007] According to one embodiment, determining the correspondence comprises supplying the extracted text item and the image including the region of the text photo showing text from which the machine learning model has extracted the text item to the generative artificial intelligence model.

[0008] According to one embodiment, at least the extracted text item is supplied to the generative artificial intelligence model in a text prompt for the generative artificial intelligence model.

[0009] According to one embodiment, the generative artificial intelligence model is a large language model supporting image input.

[0010] According to one embodiment, the method comprises generating a plurality of training data elements by, for each text item of a plurality of text items in one or more text photos, extracting the text item from a respective text photo of the one or more text photos using the machine learning model, determining a correspondence between the extracted text item and a respective image including a region of the respective text photo showing text from which the machine learning model has extracted the text item by means of the generative artificial intelligence model, generating a training data element with the extracted text item as ground truth for the respective text photo in response to the determined correspondence being above a predetermined threshold and training the machine learning model using the generated training data elements.

[0011] According to one embodiment, the method comprises correcting the extracted text item by the generative artificial intelligence model in response to the determined correspondence being below the predetermined threshold and generating the training data element with the corrected text item as ground truth for the text photo.

[0012] According to one embodiment, the method comprises rejecting usage of the extracted text item for one of the training data elements in response to the determined correspondence being below the predetermined threshold.

[0013] According to one embodiment, the method comprises, for each text item of the plurality of text items in the one or more text photos, obtaining user feedback regarding the correspondence between the extracted text item and the respective image, determining whether the user feedback regarding the correspondence confirms the determined correspondence and generating the training data element with the extracted text item as

ground truth for the respective text photo in response to the user feedback confirming the determined correspondence.

[0014] According to one embodiment, the method comprises rejecting usage of the extracted text item for one of the training data elements in response to the user feedback not confirming the determined correspondence.

[0015] According to one embodiment, the text item is a text description of a dish.

[0016] According to one embodiment, the text photo is a photo of a menu.

[0017] According to one embodiment, a method for extracting text items from a plurality of text photos is provided, comprising training the machine learning model according to the method of any one of the embodiments described above and extracting text items from the plurality of text photos by supplying the text photos to the trained machine learning model.

[0018] According to one embodiment, a server computer comprising an input interface, a memory interface and a processing unit configured to perform the method of any one of the embodiments described above is provided.

[0019] According to one embodiment, a computer program element is provided comprising program instructions, which, when executed by one or more processors, cause the one or more processors to perform the method of any one of the embodiments described above.

[0020] According to one embodiment, a computer-readable medium is provided comprising program instructions, which, when executed by one or more processors, cause the one or more processors to perform the method of any one of the embodiments described above.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] The invention will be better understood with reference to the detailed description when considered in conjunction with the non-limiting examples and the accompanying drawings, in which:

- FIG. 1 shows a communication arrangement of a marketplace system, including a smartphone and a server (computer).
- FIG. 2 illustrates the processing of input data provided by a merchant.
- FIG. 3 shows an example of a review page.

- FIG. 4 illustrates the main flow of a menu picture track, i.e. extraction of text from a photo of a menu.
- FIG. 5 illustrates a generative AI (artificial intelligence) add-on of the menu picture track.
- FIG. 6 illustrates an image extraction add-on of the menu picture track.
- FIG. 7 shows a flow diagram illustrating a method for training a machine learning model to extract text items from text photos.
- FIG. 8 shows a server computer according to an embodiment.

DETAILED DESCRIPTION

[0022] The following detailed description refers to the accompanying drawings that show, by way of illustration, specific details and embodiments in which the disclosure may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the disclosure. Other embodiments may be utilized and structural, and logical changes may be made without departing from the scope of the disclosure. The various embodiments are not necessarily mutually exclusive, as some embodiments can be combined with one or more other embodiments to form new embodiments.

[0023] Embodiments described in the context of one of the devices or methods are analogously valid for the other devices or methods. Similarly, embodiments described in the context of a device are analogously valid for a vehicle or a method, and vice-versa.

[0024] Features that are described in the context of an embodiment may correspondingly be applicable to the same or similar features in the other embodiments. Features that are described in the context of an embodiment may correspondingly be applicable to the other embodiments, even if not explicitly described in these other embodiments. Furthermore, additions and/or combinations and/or alternatives as described for a feature in the context of an embodiment may correspondingly be applicable to the same or similar feature in the other embodiments.

[0025] In the context of various embodiments, the articles “a”, “an” and “the” as used with regard to a feature or element include a reference to one or more of the features or elements.

[0026] As used herein, the term “and/or” includes any and all combinations of one or more of the associated listed items.

[0027] In the following, embodiments will be described in detail.

[0028] FIG. 1 shows a communication arrangement of a marketplace system, including a smartphone 100 and a server (computer) 106.

[0029] The smartphone 100 has a screen showing the graphical user interface (GUI) of an app for using one or more of various services, such as ordering food, which the smartphone's user has previously installed on his smartphone and has opened (i.e. started) to use the service, e.g. to order food.

[0030] The GUI 101 includes graphical user interface elements 102, 103 helping the user to use the service, e.g. a map of a vicinity of the user's position, food available in the user's vicinity (which the app may determine based on a location service, e.g. a GPS-based location service), a button for placing an order, etc.

[0031] When the user has made a selection for a service, e.g. a selection of a restaurant or an online supermarket and/or a selection of food or groceries to order, the app communicates with a server 106 of the respective service via a radio connection. The server 106 (carrying out a corresponding server program by means of a processor 107) may consult a memory 109 or a data storage 108 having information regarding the service (e.g. prices, availability, estimated time for delivery etc.) The server communicates any data relevant or requested by the user (such as estimated time for delivery) back to the smartphone 100 and the smartphone 100 displays this information on the GUI 101. The user may finally accept a service, e.g. order food. In that case, the server 106 informs the service provider 104, e.g. a restaurant or online supermarket accordingly. The server 106 may also communicate earlier with the service provider 104, e.g. for determining the estimated time for delivery.

[0032] It should be noted while the server 106 is described as a single server, its functionality, e.g. for providing a food delivery service will in practical application typically be provided by an arrangement (i.e. a server computer system) of multiple server computers (e.g. implementing a cloud service). Accordingly, the functionality described in the following provided by a server (e.g. server 106) may be understood to be provided by an arrangement of servers or server computers.

[0033] For providing a food ordering service, the server 106 in particular needs to have information about dishes offered by the various restaurants from which food may be ordered. So, this information (such as merchant menus) needs to be incorporated into the server 106 (e.g. into memory 109 or data storage 108) and needs to be kept up to date.

[0034] According to various embodiments, approaches are provided which allow efficient incorporating (i.e. registering or onboarding) and updating of information of merchant menus which allows avoiding a manual process for doing this which is cumbersome, repetitive, and time-consuming processes, in particular when there is a high number of merchants (e.g. restaurants). This allows reducing menu onboarding and updating cost, speeding up merchant onboarding and increasing merchant menu self-serve rate.

[0035] According to various embodiments, the process of menu onboarding and updating is automated allowing merchants to easily scan or record their menus using a (first) machine learning (ML) model and using a Large Language Model (LLM), i.e. a second ML model which is an LLM) to structure menu data, increasing efficiency and reducing costs associated with manual updates.

[0036] According to various embodiments, apart from the application of an LLM to structure the (first) ML model's output, one or more of the following may be provided:

- An end-to-end system to capture feedback from merchant edits to enable iterative ML model improvement (feedback loop system)
- A third ML model (which is a generative model) to assist in data augmentation/enhancement to improve the first ML model
- Photo dish extraction (from a photo of a menu having one or more dish photos) and linking with the first ML model and the third ML model (generative model). Photo dish extraction may be performed by a fourth ML model.

[0037] According to various embodiments, the server 106 allows a merchant (e.g. service provider 104) to scan their menu or record audio of their menu items. This input is then processed using the first ML model and the second model (LLM). The first ML model may for example be or include a model based on OpenAI Whisper for voice transcription, PSENet (Progressive Scale Expansion Network) and/or SVTR (Scene Text Recognition with a Single Visual Model) for text detection and recognition. The LLM classifies, maps, and structures the data (generated by the first ML model) into publishable menu formats.

[0038] FIG. 2 illustrates the processing of input data 201 provided by a merchant including a menu picture (i.e. an image (e.g. photo) including a textual description of the dishes) 202 and/or a voice recording of the merchant's menu 203 (i.e. a spoken version of the textual description of the dishes).

[0039] The first ML model applies text extraction 204 and/or voice transcription 205, respectively, on the input data to provide menu output data. For this, the first ML model may include two sub-models, one for text extraction and one for voice transcription. The menu output data generated by the first ML model is fed to the LLM 206 which aligns the menu output data with a predetermined menu data structure. The LLM 206 may be reviewed by a user, in particular the merchant, and published by the server 106 (menu publication 208), e.g. when it has been approved by the merchant.

[0040] So, according to various embodiments, there are two tracks of the automated menu onboarding and/or updating process: one is through a menu picture 202, another is through a voice recording of the menu 203.

[0041] For using the voice recording track

- 1) for generating the input 203, the merchant can for example use their mobile phone to record their voice. For example, the server 106 may provide guidance on an onboarding web page or app to have the merchant speak out their menu items one by one.
- 2) after the server 106 gets the audio (i.e. voice recording, which may include multiple items) uploaded to it by the merchant, it performs voice transcription by means of the first ML model which for example includes an Automatic Speech Recognition (ASR) model to transcript the voice to text.
- 3) the output of the first ML model (ASR model in this case), is menu output data which is in the form of raw text, ideally the text that the merchant has spoken into their mobile device, for example. The LLM 206 helps extracting the food item, category and price from the raw text to generate structured menu data. It should be noted that people speak and structure their things differently, so rule-based extraction may fail and a traditional ML models (which is learning from observed patterns) can also fail but an LLM is typically capable of performing that task. Example:

[0042] For example, the following text is the (raw text) menu output data (output by the first ML 204: “The first category is spaghetti, and here are all the items. The item name is beef spaghetti, its price is 5 dollars. The item name is black pepper pasta, the price of it is 7 dollars. The item name is tomato chicken pasta, the price is 8 dollars.”

[0043] The LLM output (structured menu data) for this example is then for example:

[
{'itemName': 'beef spaghetti',

```
'price': 5,  
'categoryName': 'spaghetti',  
'description': ''},  
{'itemName': 'black pepper pasta',  
'price': 7,  
'categoryName': 'spaghetti',  
'description': ''},  
{'itemName': 'tomato chicken pasta',  
'price': 8,  
'categoryName': 'spaghetti',  
'description': ''}  
]
```

[0044] Through prompt engineering, the LLM 206 may for example made to output JSON format data as in the example above, so that a downstream backend service can easily parse the data and fill in corresponding fields (of a web page or app GUI 101).

[0045] FIG. 3 shows an example of a review page 300 which presents the merchant with the processing result of the input 203 in the above example und gives the merchant the opportunity to edit this result.

[0046] The food items 301 of the review page are generated by the server 106 by parsing from the structured menu data generated by the LLM 206 for menu review 207. The item name, category, and price are given in this review page 300. The review page 300 closes the loop since the merchant - the menu owner - gets to review the processing output and edit, delete or further add some items. In this manner, the results of the automated menu onboarding and/or updating process, which acts as assistant to merchants, may be confirmed by the merchants. Thus, a Merchant Menu Assistant is provided (powered by AI) which can help save time and effort for merchants or other users.

[0047] Similarly to the voice recording track described above, the menu picture track also includes the flow Input 202 -> first ML Model (text extraction 204) -> LLM 206 -> menu review. Differently from the voice recording track, in the menu picture track some other components may be included to make the whole system a better loop, which can continuously increase the performance, in particular of the first ML model (i.e. text extraction 204). High quality of text extraction by the first ML model is very important to make menu picture track

work, and at the same time, it has more space for improvement when data from the real world is retrieved.

[0048] According to various embodiments, the menu picture track includes three functional components (e.g. respectively implemented by three sub-systems), a main flow, a generative AI add-on and an image extraction add-on which are described in the following.

[0049] FIG. 4 illustrates the main flow 400 of the menu picture track.

[0050] The main flow 400 contains an end-user feedback loop process that is used to iteratively improve a text extraction (OCR(optical character recognition)) model 401 (which is the first ML model in this case or a sub-model of the first ML model).

[0051] The main flow includes the following:

- 1) the merchant 402 inputs a menu photo 403, e.g. uploads a menu photo 403 (including one or more food items) to the server 106
- 2) .The text extraction model 401 processes the menu photo 403. The text extraction model 401 includes a text detection model and a text recognition model. The text extraction result 404 of this processing is recognition raw text 405 and the corresponding location 406 of the text in the image for each food item. (location information, i.e. in form of one or more bounding boxes). For example the text recognition result 404 (e.g. for one food item) is {"fried rice": [12,25,31,67]}
- 3) The text recognition result 404 is formatted (e.g. to fit an LLM text prompt) and then put as part of a prompt to the LLM 407. The purpose of the LLM 407 here is for understanding the menu photo 401 but instead of sending the menu photo 401 directly to the LLM 407 (which may take longer and have less accuracy), the text extraction model 401 is used to extract the text and location first the LLM 407 processes the result 404 based on its understanding of menus.
- 4) The output 408 of the LLM 407 is for example similar to the voice recording track and may be parsed to generate a review page as described above for the voice recording track (FIG. 3 for three food items) for presenting it to the merchant for review to form a merchant feedback loop.
- 5) When the merchant 402 reviews the output 408 (in form of the review page), when they may find that the food items have been correctly extracted from the menu photo, they may approve the food items and the approval 409 is logged in a log 410. Sometimes the merchant 401 may find a food item to have been extracted wrongly.

For example, the menu photo 401 is showing “fried rice” but the output 408 indicates “fire rice”. In that case, the merchant 401 may modify the food item to the correct one. Such a modification 411 is also logged in the log 410.

The approvals 409 and modifications 411 from the merchant 401 are joined back to the text recognition result 404, i.e. the text extraction result 404 is mapped to the merchant behaviour. For example, the text extraction result 404 for a food item is {“fire rice”: [12,25,31,67]} when actually it should be {“fried rice”: [12,25,31,67]}. This kind of feedback data can in the following be used to improve the text extraction model 401. For example, the updated data (modified food items) and approved data (approved food items) are collected in a “golden dataset” (i.e. training dataset) 412 as ground truth data (associated with the respective menu photos) and the golden dataset 412 is used to further fine-tune the text extraction model 401 to make it better and improve the accuracy for the menu picture track.

[0052] FIG. 5 illustrates the generative AI add-on 500 of the menu picture track.

[0053] The flow illustrated in FIG. 5 is connected to the main flow of FIG. 4 via the point A in figures 4 and 5.

[0054] The generative AI add-on 501 serves two purposes: first, as a way to augment additional data for training the text extraction model 401, and second as a filter to select good quality data from the merchant feedback loop.

[0055] For the purpose of using the generative AI add-on 500, a set of menu photos 501 is collected, e.g. from merchant upload, open source menus etc.

[0056] The menu photos 501 are then processed by the current version of the text extraction model 502 (corresponding to text extraction model 401) to output, as text extraction result 503, the text of food items and their corresponding locations in the menu photos 501.

[0057] For each food item, based on the location (e.g. in form of a bounding box), the menu photo 501 is cropped to the text and paired with the extracted text, see FIG. 5 illustrating an output 504 which is a pair of text and cropped image including as OCR text: Grilled Eggplant and an image saying ”Grilled Eggplant Zaalouk”.

[0058] In the present example, there is a mistake in the OCR text, which is missing the word “Zaalouk”. To detect this mistake and use it for improving the text extraction model 401, 502, this output 504 is sent to a multi-modal generative AI model 505 (such as GPT 4V),

which can take both text and images as input. The generative AI model 505 is used to grade the correspondence between the cropped image and the OCR text and/or (if applicable) correct the OCR text that is provided. Its output 506 is thus a pair of (possibly) corrected text and cropped image. Effectively, the generative AI model 505 is thus used to automate the data validation and reviewing process instead of relying on humans (thus saving effort and cost).

[0059] In the present example, the generative AI model 505 corrects “Grilled Eggplant” to “Grilled Eggplant Zaalouk” to generate a ground truth 507 for the food item (which may be added to the golden dataset 412 to further train and fine-tune the text extraction model 401, 502 or it outputs “poor correspondence” which may also be used as feedback to further train and fine-tune the text extraction model 401, 502.

[0060] For example, the training data elements of the golden dataset 412 may be used for a (re-)training round of the machine text extraction model 401, 502, e.g. by computing a loss of its output for the respective text photos and the ground truth and adjusting its parameters (e.g. neural network weights) in direction of decreasing loss.

[0061] For the purpose of filtering good quality data from the merchant feedback loop, merchant output (i.e. a modified or approved food item) is compared with the output from the generative AI model 505. Only data that has a good overlap with the generative AI model output or if the output is rated as good by the generative AI model 505 is retained for further model improvement (e.g. is put into the golden dataset 412).

[0062] FIG. 6 illustrates the image extraction add-on 600 of the menu picture track.

[0063] The flow illustrated in FIG. 6 is connected to the main flow of FIG. 4 via the points B in figures 4 and 6.

[0064] The purpose of the image extraction add-on 600 is to detect and extract an image of the dish from the menu photo 401 in case it is a menu with dish photo (i.e. shows how the dish looks like) and pairing it with the correct dish name and its corresponding price which was extracted by the text extraction model.

[0065] So, the input 601 of the image extraction add-on 600 is again a menu photo from merchant upload but assumed to include an image of the dish. An object detection model 602 is used to get the bounding box (location information) of the (or each) dish image in this menu photo 601 i.e. the object detection model 602 outputs one or more pairs 603 <dish image - location>.

[0066] Then, taking the text extraction result 604 from the main flow 400, which includes a <text - location> pair for each food item, the extracted text and dish images may be mapped one to each other, e.g. by a multi-modal generative AI model 605 (e.g. corresponding to multi-modal generative AI model 505 or the LLM 407) and/or based on the location of the food item text (i.e. name) and dish image, the dish images may be assigned to the corresponding food items to form pairs 606 of dish image and food item text. This result 606 of the image extraction add-on 600 may be sent back to the merchant 401 to review and when the merchant 401 approves dish images may be included in the presentation of the menu to customers (e.g. GUI 101). Thus, the image extraction add-on 600 helps merchants to onboard their dish photos more efficiently.

[0067] In summary, according to various embodiments, a method is provided as illustrated in FIG. 7.

[0068] FIG. 7 shows a flow diagram 700 illustrating a method for training a machine learning model to extract text items from text photos.

[0069] In 701, a text item is extracted from a text photo using the machine learning model.

[0070] In 702, a correspondence between the extracted text item and an image including a region of the text photo showing text from which the machine learning model has extracted the text item by means of a generative artificial intelligence model is determined.

[0071] In 703, the machine learning model is adjusted depending on the determined correspondence.

[0072] According to various embodiments, in other words, training data or feedback for a text extraction model is provided by means of a generative artificial intelligence model. The trained machine learning model (text extraction model) can for example applied to extract food items (in editable text form) from menu photos, i.e. to provide a menu assistant, but it may also be applied in other (e.g. industry) use cases that require cataloguing and structuring varied and extensive data from images (e.g. in addition to voice), such as restaurant booking platforms, inventory management systems in the retail sector, or any other platforms dealing with product listings and updates.

[0073] As described above, a system is for example provided that comprises

- a main flow which implements an end-to-end system to capture feedback from merchant edits to enable iterative ML model improvement (feedback loop system)

- a generative AI add-on which uses generative artificial intelligence to assist in data augmentation/enhancement to improve the ML model
- an image extraction add-on which performs photo extraction and dish name mapping by means of generative AI or ML.

[0074] The method of FIG. 7 is for example carried out by a server computer as illustrated in FIG. 8.

[0075] FIG. 8 shows a server computer 800 according to an embodiment.

[0076] The server computer 800 includes an input interface 801 (e.g. configured to receive the text photo). The server computer 800 further includes a processing unit 802 and a memory 803. The memory 803 may be used by the processing unit 802 to store, for example, data to be processed, such as information about demand and supply. The server computer is configured to perform the method of FIG. 7.

[0077] The methods described herein may be performed and the various processing or computation units and the devices and computing entities described herein (e.g. the server computer) may be implemented by one or more circuits. In an embodiment, a "circuit" may be understood as any kind of a logic implementing entity, which may be hardware, software, firmware, or any combination thereof. Thus, in an embodiment, a "circuit" may be a hard-wired logic circuit or a programmable logic circuit such as a programmable processor, e.g. a microprocessor. A "circuit" may also be software being implemented or executed by a processor, e.g. any kind of computer program, e.g. a computer program using a virtual machine code. Any other kind of implementation of the respective functions which are described herein may also be understood as a "circuit" in accordance with an alternative embodiment.

[0078] While the disclosure has been particularly shown and described with reference to specific embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention as defined by the appended claims. The scope of the invention is thus indicated by the appended claims and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced.

CLAIMS

1. A method for training a machine learning model to extract text items from text photos, comprising:
Extracting a text item from a text photo using the machine learning model;
Determining a correspondence between the extracted text item and an image including a region of the text photo showing text from which the machine learning model has extracted the text item by means of a generative artificial intelligence model; and
adjusting the machine learning model depending on the determined correspondence.
2. The method of claim 1, wherein adjusting the machine learning model depending on the determined correspondence comprises adjusting a processing of the text photo by the machine learning model depending on the determined correspondence.
3. The method of claim 1 or 2, wherein adjusting the machine learning model depending on the determined correspondence comprises adjusting the machine learning model to improve the correspondence if the correspondence is below a predetermined threshold.
4. The method of any one of claims 1 to 3, wherein determining the correspondence comprises supplying the extracted text item and the image including the region of the text photo showing text from which the machine learning model has extracted the text item to the generative artificial intelligence model.
5. The method of claim 4, wherein at least the extracted text item is supplied to the generative artificial intelligence model in a text prompt for the generative artificial intelligence model.
6. The method of any one of claims 1 to 5, wherein the generative artificial intelligence model is a large language model supporting image input.

7. The method of any one of claims 1 to 6, comprising generating a plurality of training data elements by, for each text item of a plurality of text items in one or more text photos,
 - extracting the text item from a respective text photo of the one or more text photos using the machine learning model;
 - determining a correspondence between the extracted text item and a respective image including a region of the respective text photo showing text from which the machine learning model has extracted the text item by means of the generative artificial intelligence model;
 - generating a training data element with the extracted text item as ground truth for the respective text photo in response to the determined correspondence being above a predetermined threshold; and
 - training the machine learning model using the generated training data elements.
8. The method of claim 7, comprising correcting the extracted text item by the generative artificial intelligence model in response to the determined correspondence being below the predetermined threshold and generating the training data element with the corrected text item as ground truth for the text photo.
9. The method of claim 7, comprising rejecting usage of the extracted text item for one of the training data elements in response to the determined correspondence being below the predetermined threshold.
10. The method of any one of claims 7 to 9, comprising, for each text item of the plurality of text items in the one or more text photos,
 - obtaining user feedback regarding the correspondence between the extracted text item and the respective image;
 - determining whether the user feedback regarding the correspondence confirms the determined correspondence; and
 - generating the training data element with the extracted text item as ground truth for the respective text photo in response to the user feedback confirming the determined correspondence.

11. The method of claim 10, comprising rejecting usage of the extracted text item for one of the training data elements in response to the user feedback not confirming the determined correspondence.
12. The method of any one of claims 1 to 11, wherein the text item is a text description of a dish.
13. The method of any one of claims 1 to 12, wherein the text photo is a photo of a menu.
14. A method for extracting text items from a plurality of text photos, comprising training the machine learning model according to the method of any one of claims 1 to 13; and extracting text items from the plurality of text photos by supplying the text photos to the trained machine learning model.
15. A server computer comprising an input interface, a memory interface and a processing unit configured to perform the method of any one of claims 1 to 14.
16. A computer program element comprising program instructions, which, when executed by one or more processors, cause the one or more processors to perform the method of any one of claims 1 to 14.
17. A computer-readable medium comprising program instructions, which, when executed by one or more processors, cause the one or more processors to perform the method of any one of claims 1 to 14.