Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference “Dialogue 2023”

June 14–16, 2023

Text VQA with Token Classification of Recognized Text and Rule-Based Numerical Reasoning

Surkov V. O.
Moscow Institute of Physics and Technology
Dolgoprudny, Russia
surokpro2@gmail.com

Evseev D. A.
Moscow Institute of Physics and Technology
Dolgoprudny, Russia
dmitrij.euseew@yandex.ru

Abstract

In this paper, we describe a question answering system on document images which is capable of numerical reasoning over extracted structured data. The system performs optical character recognition, detection of key attributes in text, generation of a numerical reasoning program, and its execution with the values of key attributes as operands. OCR includes the steps of bounding boxes detection and recognition of text from bounding boxes. The extraction of key attributes, such as quantity and price of goods, total etc., is based on the BERT token classification model. For expression generation we investigated the rule-based approach and the T5-base model and found that T5 is capable of generalization to expression types unseen in the training set. The proposed architecture of the question answering system utilizes the structure of independent blocks, each of which can be enhanced or replaced while keeping other components unchanged. The proposed model was evaluated in the Receipt-AVQA competition and on FUNSD dataset.

Keywords: visual question answering, optical character recognition, receipt images, token classification, numerical reasoning

DOI: 10.28995/2075-7182-2023-22-486-496

Ответ на вопросы по тексту на изображениях с помощью классификации токенов распознанного текста и численных рассуждений на основе правил

Сурков В. О.
Московский физико-технический институт
Долгопрудный, Россия
surokpro2@gmail.com

Евсеев Д. А.
Московский физико-технический институт
Долгопрудный, Россия
dmitrij.euseew@yandex.ru

Аннотация

В данной работе описывается система для ответа на вопросы по изображениям с текстом с возможностью численного рассуждения по извлеченным структурированным данным. Система выполняет распознавание текста на изображении, определение ключевых атрибутов в тексте, генерацию выражения для численного рассуждения и его выполнение с ключевыми атрибутами в качестве аргументов. Распознавание текста включает в себя следующие этапы: определение областей с текстом на изображении и последующий перевод их в текст. Извлечение ключевых атрибутов, таких как количество и цена товаров, сумма и т. д. выполняется моделью классификации токенов на основе BERT. Для генерации выражений были исследованы подход на основе правил и модель T5-base и установлено, что T5 способен к обобщению на типы выражений, не встречающиеся в обучающей выборке. Архитектура вопросно-ответной системы реализована в виде набора независимых блоков, каждый из которых может быть заменен или улучшен при сохранении остальных компонентов неизменными. Предложенная модель была применена в соревновании Receipt-AVQA и протестирована на датасете FUNSD.

Ключевые слова: ответ на вопросы по изображениям, распознавание текста, изображения товарных чеков, классификация токенов, численные рассуждения

1 Introduction

Visual Question Answering (VQA) is the task of finding an answer given an image and a question in natural language. Text VQA is the subfield of VQA which involves reading text on images such as signboards, receipts, documents etc. Answering to the questions about text on images requires performing optical character recognition and fusion of text and image representations.

One of the first approaches to VQA (Kazemi and Elqursh, 2017) was based on processing of an image with CNN, a question with RNN, attention between question and image representations and classification of possible answers. A similar approach to Text VQA (Singh et al., 2019) includes recognition of text on images and obtaining scene text representations. Pretraining of Transformers (Vaswani et al., 2017) on images and recognized text (Yang et al., 2021), (Li et al., 2021b), (Biten et al., 2022) with the objective functions of masked language modeling, masked image modeling and word-patch alignment improves the quality of question answering.

In our paper we describe the system for extraction of structured information from document images and subsequent question answering. The system can be applied to understanding receipt or form images which has many applications in industry. For example, information, extracted from receipts, is useful to keep track of customers' expenses or to optimize the supply chain of companies. The system is capable of numerical reasoning over extracted key attributes during answer generation. The question answering system includes the following components: building a numerical reasoning expression for the question, extraction of structured information from the image and execution of the expression with extracted values as operands. This pipeline-based approach enables replacement of any component for more elaborate one and makes the process of answer generation interpretable. The model was trained and evaluated on Receipt-AVQA dataset which contains receipt images, text and questions and FUNSD (Jaume et al., 2019) dataset of form images. The proposed system scored MASE of 0.1164 on QA track and 0.2331 on VQA track of Receipt-AVQA competition and achieves competitive performance (F1=78.4) on FUNSD dataset.

2 Related Work

Text VQA. Question answering on images with textual content, such as signboards, recepts, invoices etc. (Singh et al., 2019), (Biten et al., 2019), (Mishra et al., 2019), has been an active area of research in last years. TextVQA (Singh et al., 2019) is one of the first datasets which contains questions related to text on images. The authors of the dataset proposed the LoRRa model, which is based on fusion of question, image and OCR text representations, and subsequent classification on the vocabulary words. The model (Mishra et al., 2019) performs text block extraction and defines which of the blocks contains the answer. Unlike the approaches of late fusion of image and text representations, obtained with CNNs (Lin et al., 2017), (Simonyan and Zisserman, 2014) and LSTM (Hochreiter and Schmidhuber, 1997), M4C (Hu et al., 2020) is a multimodal Transformer which takes as input the embeddings of question words, detected words and OCR tokens. The answer is generated in autoregressive way with dynamic pointer network. M4C outperforms previous approaches on TextVQA dataset.

Pretraining of language models on images and recognized text leads to further improvements in the task of Text VQA, because it gives better joint representations than a sole objective toward correct answer. In Text-Aware Pre-training (Yang et al., 2021) embeddings of text words, visual objects and scene text are fed into the multi-modal Transformer, pretrained with masked-language modeling (MLM), relative position prediction and image-text matching objectives. Layout Transformer (Biten et al., 2022) is pretrained on text with spatial cues (coordinates of the text region) on denoising task. In SelfDoc (Li et al., 2021b) the Transformer takes as input sentence embeddings of the text from the document and embeddings of object proposals and is pretrained with MLM objective. ERNIE-Layout (Peng et al., 2022) adopts a reading order prediction task in pre-training and spatial-aware disentangled attention mechanism. LayoutLMv3 (Huang et al., 2022) is pretrained with unified text and image masking and word-patch alignment to learn cross-modal alignment. LayoutLMv3 achieves SOTA performance on text-centric and image-centric VQA tasks.

Sequence Tagging. In our system relevant numerical values are extracted from receipt OCR text using sequence tagging method, which involves matching categorical labels to sequence items. Its classical examples are Part-of-speech tagging and Named Entity Recognition. The common approach to sequence tagging involves encoding of text tokens with BiLSTM (Lample et al., 2016), CNN (Ma and Hovy, 2016) or pretrained language models (Devlin et al., 2018a) (Bao et al., 2020) and subsequent classification of hidden states or Conditional Random Field layer (Lafferty et al., 2001).

Generation of expressions for numerical reasoning Questions in Receipt-AVQA require numerical reasoning, which is commonly performed with encoder-decoder architecture. ELASTIC (Zhang and Moshfeghi, 2022) encodes a task text with RoBERTa (Liu et al., 2019) and separately generates operators and operands for the final mathematical expression. In the work of (Cobbe et al., 2021) GPT3 (Brown et al., 2020) models generate a chain of reasoning and verify it to validate reasoning correctness.

3 Task and data

Receipt-AVQA is a question answering task that requires answering a quantitative question related to a given receipt instance. The task comprises two tracks: Visual Question Answering and Question Answering. In the VQA track, the receipts' instances are provided as images, while in the QA track, participants are given all text tokens from receipts along with their coordinates.

There are three types of questions: *amount*, *count*, and *ratio*, which denote the expected answer type. Each receipt uses one of two currencies: *Malaysian ringgit* and *Indonesian rupiah*, which have different scales. Participants have access to question types and currencies, as well as lists of operations required to calculate the answer (e.g., subtraction, sorting).

The participants' solutions are evaluated using the metric, based on MASE score. Specifically, all questions are divided into six groups based on question type and currency, and MASE values are calculated for each group, the scores are then averaged. The task and evaluation method pose difficulties, as answers can lie in a wide range.

The dataset consists of 21,837 questions (16,611 in the training subset, 2,302 in the development subset, and 2,924 in the testing subset) and 1,957 receipts (1,537 in the training subset, 210 in the development subset, and 210 in the testing subset).

4 Method

The scheme of the proposed approach is depicted in Figure 1. Optical character recognition (4.1) is used to transform a photo of a receipt into textual information, thereby reducing the VQA task to a QA task. The Attribute Extractor (4.3) extracts numerical information and structures it. The Question Processor (4.2) accepts a question and generates a mathematical expression. Finally, the Answer Generator (4.4) produces an answer from the receipt contents (constructed by the Attribute Extractor) and the expression.

4.1 Image-to-Text Extraction

Text recognition in our model is performed in the following steps:

- Detection of regions with text on the image;
- Cropping the text regions and feeding them into the model, which generates text;
- Splitting text regions into lines and sorting by the line number from top to bottom and within the line from left to right (using the coordinates of detected text regions).

The text detection component utilizes the PP-OCRv3 (Li et al., 2022) architecture. PP-OCRv3 includes the Path Aggregation Network (PAN)(Liu et al., 2018) for the calculation of feature maps and the Feature Pyramid Network(Lin et al., 2017) for object detection (regions on the image with text in our case). We trained PP-OCRv3 on bounding boxes with the text from the train split of the Receipt-AVQA-2023 dataset. The model was trained in three epochs with a batch size of 8, learning rate of 0.001, and achieved precision=0.899, recall=0.905 on the dev split. An example of detected text regions can be seen in Figure 2. Text recognition in detected regions is based on the Transformer encoder-decoder TrOCR model (Li et al., 2021a).

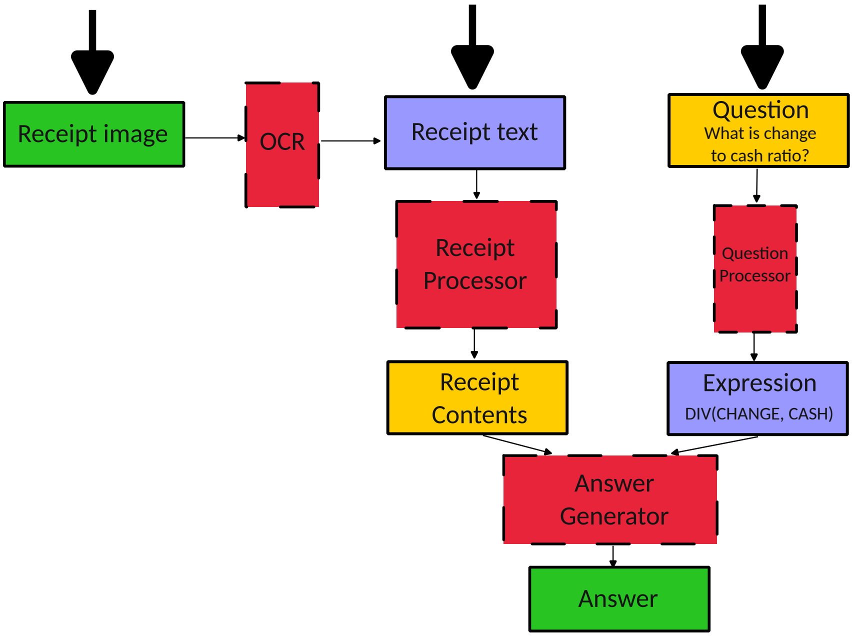
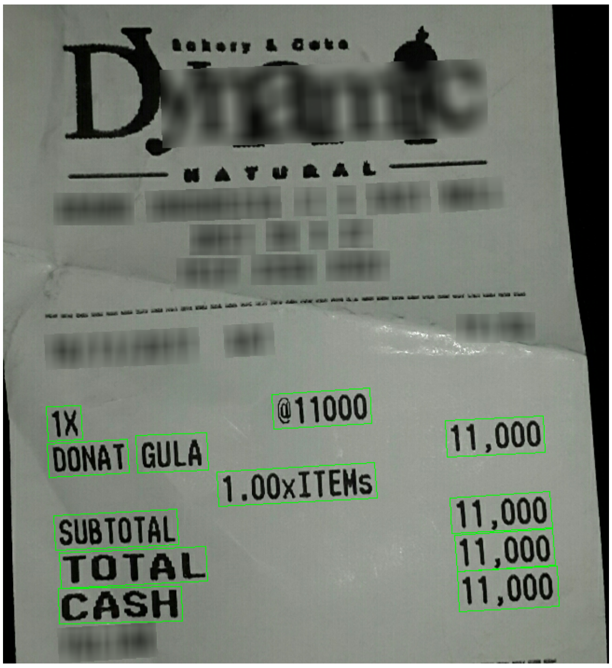


Figure 1: Model scheme



1X @11000
DONAT GULA 11,000
1.00xITEMs
SUBTOTAL 11,000
TOTAL 11,000
CASH 11,000

Figure 2: An example of text detection using PP-OCRv3.

4.2 Question Processor

Question Processor transforms questions in English into mathematical expressions. Expressions provide exhaustive information on how to generate an answer provided all variables. We analyzed two approaches to question processing: a rule-based approach and a generative model. Description of expression structure can be found in Appendix A.1.

4.2.1 Rule-Based Question Processing

We divided questions into 50 groups, each with its own expression. To figure out which group a question belongs to, each group of questions is matched against a regular expression, which represents the group. For instance, questions «What is the average price of a position?» and «What is the mean price of a position?» belong to the same group with expression `DIV(SUM(AMOUNTS),COUNT(AMOUNTS))` and regex `'What is the (average|mean) price of a position?'`. Then, all numbers from the question are extracted, and they will be used to substitute `NUM1` and `NUM2` later (if `NUM1` and `NUM2` are needed).

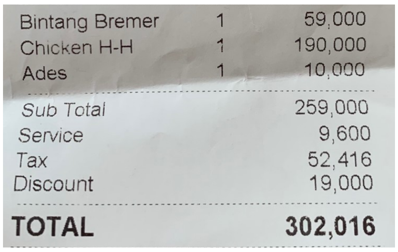
This approach is sufficient for the competition as participants have access to questions and new types of questions can be added manually.

4.2.2 T5 Question Processing

Since the rule-based approach does not generalize to new questions, we decided to develop an approach based on a generative model. We generated expressions for all questions from the train subset and fine-tuned T5-base (Raffel et al., 2020) to yield an expression given a question. The T5-base was trained on 1 epoch with a batch size of 32, AdamW optimizer (Loshchilov and Hutter, 2017), a learning rate of 1.5×10^{-4}, weight decay of 0.01, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$.

4.3 Attribute Extractor

The purpose of Attribute Extractor is extracting and structuring necessary numerical data of a receipt. This structured data is referred to as *Receipt Contents* in the model scheme 1. For each receipt we keep its general values (e. g. total, tax), and for each good we keep its unit price, quantity and total price. The example of receipt contents is shown in Figure 3.



Bintang Bremer	1	59,000
Chicken H-H	1	190,000
Ades	1	10,000
Sub Total		259,000
Service		9,600
Tax		52,416
Discount		19,000
TOTAL		302,016

```
{
  "card": null,
  "cash": null,
  "change": null,
  "discount": 19000.0,
  "goods": [
    {
      "price": 59000.0,
      "quantity": 1.0,
      "total": 59000.0
    },
    {
      "price": 190000.0,
      "quantity": 1.0,
      "total": 190000.0
    },
    {
      "price": 10000.0,
      "quantity": 1.0,
      "total": 10000.0
    }
  ],
  "round": null,
  "service": 9600.0,
  "subtotal": null,
  "tax": 52416.0,
  "total": 302016.0
}
```

Figure 3: `train/receipt_00003` image and corresponding contents in json format

4.3.1 Line Breaking

The textual information of a receipt comprises a set of words along with the coordinates of the rectangles containing them. To facilitate further text processing, the set of rectangles is divided into subsets of lines. A greedy algorithm is used for line splitting, which prioritizes pairs of rectangles with large intersections. Since the receipt is split into lines, coordinates are no longer required.

4.3.2 Rule-Based Approach

Given a sequence of receipt lines, a rule-based attribute extractor produces structured information about the receipt. The algorithm is divided into two parts: parsing goods and parsing general information.

In the first part, the rule-based attribute extractor creates a list of goods by searching for the unit price, quantity, and total price for each position. In the second part, the extractor finds general values (such as change or service fee), more details are given in Appendix A.2.

However, this approach has several flaws. Firstly, the set of strategies is not exhaustive and the model cannot handle novel formats of goods. Secondly, it cannot handle receipts with a non-unified format of goods. Lastly, it cannot parse lines containing two or more general values.

4.3.3 BERT Approach

First, we describe the process of constructing the training dataset for our BERT approach. We used the rule-based method mentioned above to generate receipt contents for the training subset. We considered the rule-based approach to have produced the correct receipt contents for a receipt if, when using this content, the entire model produced the correct answers to all questions for that receipt. We then ruled out incorrect receipts, and this formed the training dataset for our BERT approach.

To generate receipt contents, we used two BERT-base models (Devlin et al., 2018b) referred to as $BERT_{labels}$ and $BERT_{goods}$. Both models were used for a tagging problem, where the sequence to be tagged is the concatenation of a receipt's lines. $BERT_{labels}$ predicted tokens containing general values (e.g., tokens `B-TOTAL` and `O-TOTAL`) or information about a particular good (tokens `B-POSITION` and `O-POSITION`). Similarly, $BERT_{goods}$ predicted tokens containing values related to goods (e.g., tokens `B-PRICE` and `O-PRICE`). We adjusted the rule-based approach to yield these tags, and using all information about the tags, we could unambiguously identify all general values and a list of goods, and form the receipt contents.

Both BERT models were trained for 30 epochs with a batch size of 20, using the AdamW optimizer (Loshchilov and Hutter, 2017), a learning rate of 2×10^{-5}, weight decay of 10^{-6}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-6}$.

With tags obtained from $BERT_{labels}$ and $BERT_{goods}$, we identified the tokens containing numbers for the values of the receipts. To extract a number from a token, we removed any letters and other symbols unrelated to the number and replaced the decimal point with a comma if it was represented as a comma.

4.3.4 Pruning

The MASE metric highly penalizes large errors, even in a small part of the sample, unlike the accuracy metric. Therefore, we decided to prune large answers. Specifically, for each pair *(currency, expression)*, we calculated the median m, average a, and standard deviation s on the corresponding subset of the training dataset. If an answer exceeded $a + 3s$, we replaced it with m. We chose the median as the replacement value because it minimizes the MAE.

4.4 Answer Generator

The answer generator uses an expression based on the question and receipt contents to yield an answer to the task. First, any missing information on the receipt contents is filled in. For example, if there is no information about the unit price of a position in a receipt, it is calculated by dividing the total price by the quantity. After that, all variables in the expression are replaced with their respective values from the receipt contents. The resulting expression consists only of procedures and numbers, which are then evaluated. The final value obtained from evaluating this expression is the answer to the task.

5 Experiments and Analysis

5.1 Results on Receipt-AVQA dataset

The T5 model for question processing achieved an absolute quality score of 100% on both the development and test subsets, indicating that its performance was flawless and there were no errors or inaccuracies in its processing of the questions. To explore what questions the model can handle and to what extent it can generalize, we tested it against a pre-prepared list of questions. The results are presented in Table 4.

The model sometimes succeeds in generating correct expressions for reformulated questions and unknown expressions, but it is not reliable for very complex novel structures and wordings as it tends to imitate known expressions.

The rule-based approach for receipt processing generated all correct answers for 68% of the 1041 receipts in the train subset and for 70% of the 147 receipts in the development subset. These receipts were used as the training and validation datasets for the BERT approach. The results of both approaches are presented in Table 1.

Model	Total	Amount	Count	Ratio	Accuracy 10%
Rule-Based	0.2230	0.1338	0.3707	0.1645	84.99%
BERT	**0.1164**	**0.0844**	**0.1020**	**0.1627**	**91.45%**
OCR+Rule-Based	0.3073	**0.2952**	0.4106	0.2161	75.41%
OCR+BERT	**0.2331**	0.3427	**0.1573**	**0.1994**	**81.91%**

Table 1: Results on the test set of Receipt-AVQA (MASE metrics)

The BERT approach outperforms the rule-based one in almost all metrics. This showcases that BERT is able to generalize its knowledge about the structure and contents of receipts and overcome some of the disadvantages of the rule-based approach.

Additionally, we provided the time and memory performance of some components A.3.

5.2 Results on FUNSD dataset

The pipeline of our model can be applied to structured information extraction from any kind of document images (not only receipts). Transformer-based Attribute Extractor component was trained and tested on FUNSD dataset, which contains form images and corresponding annotations: recognized text, coordinates of regions with text and tags of entities ("header", "question", "answer", "other"). The coordinates of the boxes (text regions) were used to split the boxes list into lines. The special token "<ln>" was inserted at the beginning of each line, the special token "<box>" – at the beginning of each box.

BERT-base model was replaced by Longformer-base (Beltagy et al., 2020) to enable processing of long texts in forms. The model was trained for 30 epochs with a batch size of 20, using the AdamW optimizer (Loshchilov and Hutter, 2017), a learning rate of 2×10^{-5}, weight decay of 10^{-6}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-6}$. Attribute Extractor achieves competitive performance (F1=78.4) on FUNSD dataset (Table 2).

Model	F1
UniLMv2-base (Bao et al., 2020)	68.9
UniLMv2-large (Bao et al., 2020)	72.6
Our model	78.4
LayoutLMv2-base (Xu et al., 2020)	82.8
LayoutLMv3-large (Huang et al., 2022)	92.1
ERNIE-Layout-large (Peng et al., 2022)	93.1

Table 2: Results on FUNSD dataset

6 Conclusions

In this paper, we present a question answering system on document images which is capable of numerical reasoning over extracted structured data. The proposed architecture of the question answering system utilizes the structure of independent blocks, each of which can be enhanced or replaced while keeping other components unchanged. The system includes the following components: OCR, Attribute Extractor, which finds values of key attributes in text, Question Processor, which defines a numerical reasoning expression, and Answer Generator. Text recognition is performed using the TrOCR model which generates text from bounding boxes detected by PP-OCRv3. The Attribute Extractor is based on BERT for token classification. In the Answer Generator component we applied a rule-based approach and a T5-based model.

The proposed model achieves competitive performance on FUNSD dataset. Also, the model was evaluated in the Receipt-AVQA competition, the version with the BERT receipt processor scored MASE of 0.1164 on the QA track and MASE of 0.2331 on the VQA track. Additionally, while this is not reflected in the competition score, we found that T5 is capable of generalization to expression types unseen in the training set, making the whole scheme more resilient to new question types.

References

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, et al. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. // *International conference on machine learning*, P 642–652. PMLR.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150.*

Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. 2019. Scene text visual question answering. // *Proceedings of the IEEE/CVF international conference on computer vision*, P 4291–4301.

Ali Furkan Biten, Ron Litman, Yusheng Xie, Srikar Appalaraju, and R Manmatha. 2022. Latr: Layout-aware transformer for scene-text vqa. // *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, P 16548–16558.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168.*

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018a. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805.*

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018b. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. 2020. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. // *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, P 9992–10002.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. // *Proceedings of the 30th ACM International Conference on Multimedia*, P 4083–4091.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. Funsd: A dataset for form understanding in noisy scanned documents. // *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, P 1–6. IEEE.

Vahid Kazemi and Ali Elqursh. 2017. Show, ask, attend, and answer: A strong baseline for visual question answering. *arXiv preprint arXiv:1704.03162.*

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360.*

Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. 2021a. Trocr: Transformer-based optical character recognition with pre-trained models. *arXiv preprint arXiv:2109.10282.*

Peizhao Li, Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Varun Manjunatha, and Hongfu Liu. 2021b. Selfdoc: Self-supervised document representation learning. // *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, P 5652–5660.

Chenxia Li, Weiwei Liu, Ruoyu Guo, Xiaoting Yin, Kaitao Jiang, Yongkun Du, Yuning Du, Lingfeng Zhu, Baohua Lai, Xiaoguang Hu, et al. 2022. Pp-ocrv3: More attempts for the improvement of ultra lightweight ocr system. *arXiv preprint arXiv:2206.03001*.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. // *Proceedings of the IEEE conference on computer vision and pattern recognition*, P 2117–2125.

Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. 2018. Path aggregation network for instance segmentation. // *Proceedings of the IEEE conference on computer vision and pattern recognition*, P 8759–8768.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. Ocr-vqa: Visual question answering by reading text in images. // *2019 international conference on document analysis and recognition (ICDAR)*, P 947–952. IEEE.

Qiming Peng, Yinxu Pan, Wenjin Wang, Bin Luo, Zhenyu Zhang, Zhengjie Huang, Teng Hu, Weichong Yin, Yongfeng Chen, Yin Zhang, et al. 2022. Ernie-layout: Layout knowledge enhanced pre-training for visually-rich document understanding. *arXiv preprint arXiv:2210.06155*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. // *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, P 8317–8326.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. 2020. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*.

Zhengyuan Yang, Yijuan Lu, Jianfeng Wang, Xi Yin, Dinei Florencio, Lijuan Wang, Cha Zhang, Lei Zhang, and Jiebo Luo. 2021. Tap: Text-aware pre-training for text-vqa and text-caption. // *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, P 8751–8761.

Jiaxin Zhang and Yashar Moshfeghi. 2022. Elastic: Numerical reasoning with adaptive symbolic compiler.

A Appendix

A.1 Question Processor metadata

An expression consists of variables and procedures. A variable can denote a number (e.g., `TOTAL` and `CHANGE` account for the total amount of purchase and change, respectively) or a list of numbers (e.g., `PRICES` designates a list of unit prices of goods in the same order as they appear in the receipt). There are two special variables `NUM1` and `NUM2` for the first and the second number in a question. A procedure designates an operation that should be performed on its arguments. For example, the expression `SUM(FIRST_POSITIONS(NUM1,PRICES))` means «get the first `NUM1` elements of the `PRICES` list and calculate their sum». The lists of variables and procedures are presented in Table 3.

Variable	Explanation
TOTAL	Total amount of purchase
SUBTOT	Total excluding taxes. Used if it is explicit in the receipt. May not coincide with the subtotal inscription in the receipt
CASH	Cash used for payment
CARD	Payment by card
TAX	Tax amount
CHANGE	Change amount
DISCOUNT	Amount discounted
ROUND	Rounding value
SERVICE	Service fee
PRICES	List of unit prices (the same order as in a receipt)
QUANTITIES	List of quantities (the same order as in a receipt)
AMOUNTS	List of total prices of positions (the same order as in a receipt)
NUM1	First number in the question
NUM2	Second number in the question

Procedure	Explanation
ADD	$a+b$
SUB	$a-b$
MUL	ab
DIV	$\frac{a}{b}$
INTDIV	$[\frac{a}{b}]$
COUNT	List length
FIRST_POSITIONS	First n elements of a list
IS_ZERO	$I\{L_i = 0\}$
MIN, MAX, SUM	Minimum, maximum element or sum of elements of a list
AMIN, AMAX	Position of a minimum or maximum element
LARGER_THAN, SMALLER_THAN	Only values larger (smaller) than threshold remain in a list
LARGER_EQ_THAN, SMALLER_EQ_THAN	Only values larger (smaller) or equal to threshold remain in a list

Table 3: Lists of variables and procedures

A.2 Rule-based approach to numerical data extraction from receipts

At the first stage, the attribute extractor searches the unit price, quantity, and total price for each position. As different receipts can have different positions, a set of strategies has been developed for finding the positions of the quantities. For example, one of the strategies detects a good in a line if the line has the format [QUANTITY] [PRICE] [NAME OF PRODUCT] [TOTAL] [OPTIONAL TAIL], where each expression in brackets represents a substring in the line. Some strategies handle goods that occupy two or three lines. The processor iterates over the strategies from more to less specific, and attempts to find goods using the strategy. If it finds one or more goods, it terminates the iteration and yields parsed goods.

At the second stage, the processor finds general values (such as change or service fee). For each such value, it iterates over lines and determines whether the lines contain the value. If so, it returns the parsed quantity. If a line is already occupied by another value or good, it is not considered.

A.3 Model Performance

We measured average time for processing one instance of dataset and GPU memory consumption for TrOCR, Receipt Processor based on BERT and Question generator based on T5. The results are present in the Table 5.

Question	Generated expression	In train	Correct	Comment
How much is tax to total amount ratio?	`DIV(TAX,TOTAL)`	Yes	Yes	-
Divide tax to total amount, please?	`DIV(TAX,TOTAL)`	No	Yes	The same task with different wording
How much is tax divided by total?	`TAX`	No	No	Failed to handle different wording
What is the change to cash ratio?	`DIV(CHANGE,CASH)`	Yes	Yes	-
What is the cash to change ratio?	`DIV(CASH,CHANGE)`	No	Yes	Successfully dealt with argument order
How many goods in the 2nd position can be bought for 300?	`INTDIV(NUM2,FROM_TOP(NUM1,PRICES))`	Yes	Yes	-
With 300, How many goods in the 2nd position can be bought?	`INTDIV(NUM2,FROM_TOP(NUM1,PRICES))`	No	No	Should have swapped `NUM1` and `NUM2`
What is the ratio between price of the 1st position and price of the 2nd position?	`DIV(FROM_TOP(NUM1,PRICES),NUM2)`	No	No	Tried to mimic to the format of the known expressions
What is the sum of prices of the first 4 positions?	`SUM(FIRST_POSITIONS(NUM1,AMOUNTS))`	Yes	Yes	-
What is the sum of prices of the last 4 positions?	`SUM(LAST_POSITIONS(NUM1,AMOUNTS))`	No	Yes	Invented the unknown `LAST_POSITIONS` operator
How many positions are with price lower than 300?	`COUNT(SMALLER_THAN(NUM1,PRICES))`	Yes	Yes	-

What is the average price of positions with prices lower than 300?	`DIV(SUM(SMALLER_THAN(NUM1,PRICES)),`
	`COUNT(SMALLER_THAN(NUM1,PRICES))`
		No	Yes	Managed to combine known operators for a novel query

Table 4: Examples for expression generation with T5. *In train* column indicates whether the question (possibly, with other numbers) is present in the train set.

Component	GPU	Instances per second	GPU memory
TrOCR	Tesla P100	0.25	8.5 Gb
BERT Receipt Processor	Tesla P100	248	4.8 Gb
T5 Question Generator	Tesla P100	6.3	1.9 Gb

Table 5: Model components performance