Post-OCR parsing: building simple and robust parser via BIO tagging

Wonseok Hwang **Seonghyeon Kim** **Minjoon Seo** **Jinyeong Yim** **Seunghyun Park**

Sungrae Park **Junyeop Lee** **Bado Lee** **Hwalsuk Lee**

Clova AI, NAVER Corp.
{wonseok.hwang, kim.seonghyeon, minjoon.seo, jinyeong.yim, seung.park,
sungrae.park, junyeop.lee, bado.lee, hwalsuk.lee}@navercorp.com

Abstract

Parsing textual information embedded in images is important for various downstream tasks. However, many previously developed parsers are limited to handling the information presented in one dimensional sequence format. Here, we present POST OCR TAGGING BASED PARSER (POT), a simple and robust parser that can parse visually embedded texts by BIO-tagging the output of optical character recognition (OCR) task. Our shallow parsing approach enables building robust neural parser with less than a thousand labeled data. POT is validated on receipt and namecard parsing tasks.

1 Introduction

Human knowledge is often carried by natural language. To extract essential information from the textual data for various downstream tasks, it is often necessary to structuralize the data. This process is called "parsing".

Although various state-of-the-art parsers have been developed, they are all specialized in processing texts presented in one dimensional sequence format. However, in the era of smartphone, useful textual information is often visually embedded in images calling for a new type of parser.

Here, we present POST OCR TAGGING BASED PARSER (POT), a simple yet robust post-OCR parser that can structuralize textual information presented in images by BIO-tagging text segments extracted from OCR task. We validate our results over two parsing tasks: (1) receipt, and (2) namecard.

2 Model

We parse visually embedded texts in the following four separate steps (Fig. 1). First, text segments and their coordinates in images are extracted using OCR system. Next, using the coordinate information, the text segments are serialized to mimic conventional text format. Then the serialized segments are BIO-tagged. Finally, to generate final parses, the segments are grouped and combined. The detail of each step is explained below.

2.1 Optical Character Recognition

To extract visually embedded texts from an image, we used our in-house OCR system consisting of CRAFT text detector (Baek et al., 2019b) and Comb.best text recognizer (Baek et al., 2019a).

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

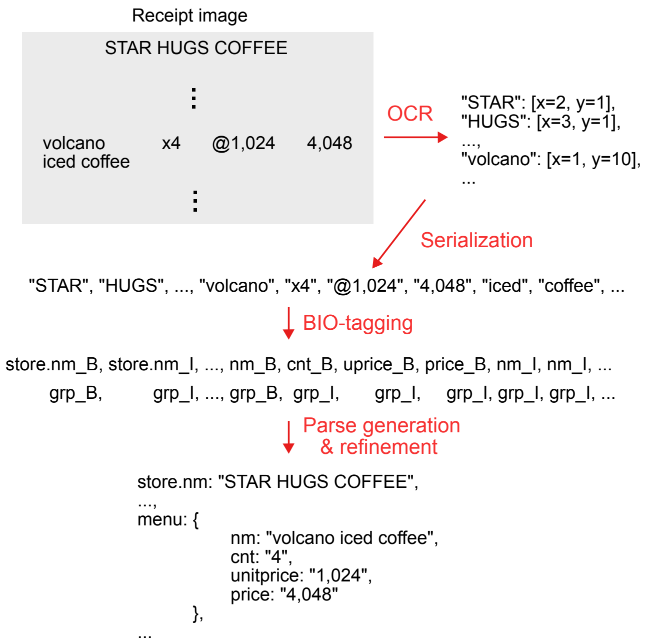
Receipt image
STAR HUGS COFFEE
volcano iced coffee
x4
@1,024
4,048
OCR
"STAR": [x=2, y=1],
"HUGS": [x=3, y=1],
...,
"volcano": [x=1, y=10],
...
Serialization
"STAR", "HUGS", ..., "volcano", "x4", "@1,024", "4,048", "iced", "coffee", ...
BIO-tagging
store.nm_B, store.nm_I, ..., nm_B, cnt_B, uprice_B, price_B, nm_I, nm_I, ...
grp_B, grp_I, ..., grp_B, grp_I, grp_I, grp_I, grp_I, grp_I, ...
Parse generation & refinement
store.nm: "STAR HUGS COFFEE",
...,
menu: {
nm: "volcano iced coffee",
cnt: "4",
unitprice: "1,024",
price: "4,048"
},
...

Figure 1: The scheme of post-OCR parsing task.

The OCR models are finetuned on each parsing dataset. Resulting text segments and their spatial information on image were delivered to the serializer.

2.2 Serialization

To maximally recover contextual information, extracted text segments are serialized according to their coordinate information as below. The algorithm first uses lexical sort to rearrange the text segments according to their coordinates from top to down and left to right direction using y axis as a primary order. Next, to group the text segments placed on the same line in the image, two segments whose the height difference is smaller than certain threshold are labeled as the same line and the segments are rearranged accordingly. When the text segments are placed on a curved line, which can happens when textual information on curled object is embedded in the image, the following line grouping algorithm is used. First, the polygon box containing text segment is moved horizontally along the angle of the text segment. If the box is overlapped with another box and their intersection area becomes higher than the pre-defined threshold, two boxes are merged into the same group and the angle is replaced by that of the new box. The process is repeated for the remaining boxes.

2.3 BIO Tagging

The serialized text segments are BIO-tagged using a neural network in following order. First, the text segments are tokenized and mapped to input vectors by adding token-, segment-, (sequential) position-, coordinate-, and line group-embeddings. The first three embeddings are prepared in identical way as BERT (Devlin et al., 2018; Vaswani et al., 2017). The segment id is set to 1 for tokens from text segments and to 0 for BERT special tokens like `[CLS]`. The coordinate embedding represents the spatial information of visually embedded text segments. Before embedding, coordinates are normalized by using image width and height and mapped to integers between 0 to 30. The line group embedding is prepared by embedding line number found in the serialization process. To tag each token, the first N_{tag} scalars of each BERT-encoded token vector are interpreted as logits where N_{tag} stands for the number of possible tags.

2.4 Parse generation

The output of the tagging model consists of mixed BIO tags of tokens (Fig. 1) which can be decoded sequentially in linear time into raw parses by noticing that for each target field, `B` and `I` of other fields can be considered as `O`-tag. In receipt parsing task, there is an additional group tag (not to be confused with line group) to reflect the hierarchical structure of parses (for example fields such as `name`, `count`, and `price` are grouped together based on the item they represent). The group BIO-tags are also decoded sequentially.

2.5 Refinements

Final parses are generated by refining raw parses. The refinement process typically involves database match and string conversion using a regular expression. In receipt parsing task, (1) various special symbols in `cnt` and `price` values, and (2) the thousands separator in `price` are refined to have unified representation. In namecard parsing task, a person's first name and family name are distinguished by using the name database and the output formats for phone and fax numbers are unified.

3 Dataset

The task of parsing post-OCR outputs using neural network has not been studied actively and we could not find appropriate datasets. Thus, we made new strong supervision datasets by ourselves. In receipt parsing task, the following information is labeled: store information, menu, and payment information. Among them, the menu consists of several grouped subfields: the name of the menu, unit price, total price, and sub-menu. The sub-menu also has a hierarchical substructure. Since these structures are not easily perceptible, we had to run pilot annotation by ourselves with 200 samples to define the ontology. Based on it, two annotation guides were created for (1) OCR data annotation, and (2) the parse tag annotation.

The annotation task was performed by crowd through in-house web application based on a centralized database. To minimize annotation errors, the crowds were assigned two roles: annotation and inspection. The annotators could start parse tag annotation only if their OCR-annotation task passed the inspection. The web application was implemented with Vue.js and Ruby on Rails. The namecard dataset was prepared similarly.

In each parsing task, train set, dev set (validation set), and test set consist of 800, 100, and 100 annotated examples. [1]

4 Experiments

The tagging model is trained via cross-entropy function with label smoothing. ADAM optimizer is used with learning rate 2e-5 with default hyperparameters. The batch size is set to 16. In receipt parsing task, tokens are augmented by randomly deleting or inserting a single token with 3.3% probability for each. Also, we attach one or two random tokens at the end of the text segment with 1.7% probability for each. Newly added tokens are randomly selected from the collection of all tokens from the train set.

5 Results & evaluation

POT is validated on two parsing tasks: receipt and namecard. To prepare evaluation metric, the oracle parses were generated from ground truth text segments. Then F_1 and sample accuracy (`acc`) were measured by comparing predicted parses to the oracle. The sample accuracy indicates the percent of samples of which all parses are predicted correctly.

The baseline model using fine-tuned multi-lingual BERT$_{\text{BASE}}$ (Devlin et al., 2018) shows $F_1 = 84.9$ (Table. 1, 1st row). Based on it, we pushed the parsing accuracy up to 90% with following improvements: (1) integration of coordinate and line-group information (`crd`, 2nd row), (2) data augmentation by randomly replacing tokens (`aug`, 3rd row), (3) advanced line-grouping for text segments placed on curved lines (`lgrp`, 4th row), (4) parse refinement (`rfn`, 5th row), and (5) loosening metric by considering predicted menu name is correct if the word-edit-distance to the oracle menu name is ≤ 2 or $\leq$ `0.4` $\times$ `length(manu names)`. The F_1 scores of individual fields

[1] The part of the datasets (or similar datasets that are free of confidential issue) will be released at `https://github.com/clovaai/cord` once the internal open-sourcing process is finished.

Table 1: The parsing accuracy table. `crd`, `aug`, `lgrp`, and `rfn` stand for "coordinate and line group information (sec. 2.3)", "data augmentation (sec. 4)", "improved line grouping (sec. 2.2)", and "output format regularization (sec. 2.5)". In `rfn2`, predicted menu name is considered as correct when word-edit-distance to the oracle is ≤ 2 or $\leq 0.4 \times$ `word length`.

		dev		test	
Model	Task	F_1	`acc`	F_1	`acc`
Baseline (BERT)	receipt	84.9 ± 0.4	19.7 ± 0.5	78.9 ± 0.7	20.0 ± 0.8
(+) `crd`	receipt	85.5 ± 0.4	22.0 ± 0.0	79.2 ± 0.7	20.3 ± 1.3
(+) `crd`, `aug`	receipt	85.8 ± 0.1	25.0 ± 1.6	79.4 ± 0.3	21.3 ± 0.5
(+) `crd`, `aug`, `lgrp`	receipt	86.9 ± 0.2	27.0 ± 1.6	79.9 ± 0.2	24.0 ± 0.0
(+) `crd`, `aug`, `lgrp`, `rfn`	receipt	89.4 ± 0.2	30.3 ± 1.3	84.7 ± 0.2	30.0 ± 0.8
(+) `crd`, `aug`, `lgrp`, `rfn2` (=POT)	receipt	91.6 ± 0.3	40.7 ± 2.1	87.2 ± 0.2	39.3 ± 1.3
POT	namecard	83.2 ± 0.8	24.0 ± 2.5	83.1 ± 0.6	23.3 ± 0.5

Table 2: The accuracies of individual fields in receipt parsing task. The dev set F_1 scores of 13 fields among 32 fields are shown.

Field	F1	Field	F1	Field	F1	Field	F1
store-info.nm	83.9	store-info.tel	80.4	store-info.address	67.7	-	-
menu.nm	97.7	menu.cnt	97.6	menu.unitprice	96.1	menu.price	97.3
pym-info.date	89.4	pym-info.time	94.6	-	-	-	-
total.total-price	98.0	total.cashprice	96.9	total.creditcardprice	80.0	total.changeprice	98.4

are shown in Table 2. Despite of the small number of supervision example (800), POT shows $F_1 = 87.2$ in the test set. To test the general applicability of POT, we also performed namecard parsing task using an identical model. POT shows F_1 = 83.1 in the test set without task-specific optimization. The relatively low F_1 compared to receipt task may be originated from a larger spatial degree of freedom in text alignment on a namecard.

6 Conclusion

We have developed a new kind of parser that can structuralize visually embedded textual information. Our shallow parsing approach based on BIO-tagging enables building robust parser with less than a thousand examples. Currently, the whole process consists of four separate modules (OCR, serialization, BIO-tagging, and parse generation) and final accuracy depends on the performance of each module. The "deep" parsing model that unifies separate processes in end-to-end fashion is currently under active development. This unified approach will allow information propagation in abstract space between modules and remove a burden of hand-crafting.

Acknowledgments

We thank Seung Shin and Young Min Baek for the help in OCR task, JeongHun Baek for the help in refinement task.

References

Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. 2019a. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *Proceedings of the IEEE International Conference on Computer Vision*.

Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. 2019b. Character region awareness for text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9365–9374.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.