Extracting Zero-shot Structured Information from Form-like Documents: Pretraining with Keys and Triggers

Rongyu Cao[1,2], **Ping Luo**[1,2,3]
[1]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing 100190, China.
[2]University of Chinese Academy of Sciences, Beijing 100049, China.
[3]Peng Cheng Laboratory, Shenzhen, China.
{caorongyu19b, luop}@ict.ac.cn

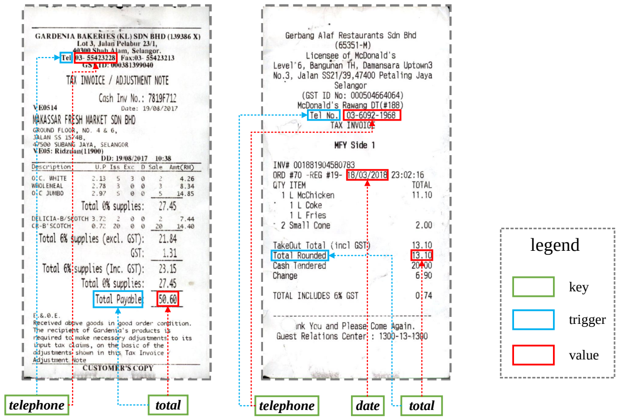
Abstract

In this paper, we revisit the problem of extracting the values of a given set of key fields from form-like documents. It is the vital step to support many downstream applications, such as knowledge base construction, question answering, document comprehension and so on. Previous studies ignore the semantics of the given keys by considering them only as the class labels, and thus might be incapable to handle *zero-shot* keys. Meanwhile, although these models often leverage the *attention mechanism*, the learned features might not reflect the *true* proxy of explanations on why humans would recognize the value for the key, and thus could not well generalize to new documents. To address these issues, we propose a Key-Aware and Trigger-Aware (KATA) extraction model. With the input key, it explicitly learns two mappings, namely from key representations to trigger representations and then from trigger representations to values. These two mappings might be intrinsic and invariant across different keys and documents. With a large training set automatically constructed based on the Wikipedia data, we pre-train these two mappings. Experiments with the fine-tuning step to two applications show that the proposed model achieves more than 70% accuracy for the extraction of zero-shot keys while previous methods all fail.

Introduction

Recent years have witnessed an increasing interest in extracting structured information from form-like documents in various vertical domains, such as invoices, purchase orders, tax forms, etc. [Zhao, Wu, and Wang 2019; Lin et al. 2020; Yu et al. 2019]. In this paper, we revisit the problem of Key Information Extraction (KIE), namely extracting the values of a set of keys from given documents [Huang et al. 2019]. For example, in Figure 1 given a set of keys ("telephone", "total") and the left receipt document, KIE task aims to extract the value "03-55423228" for "telephone" and the value "50.60" for "total". The extracted structured information is essential for a wide range of downstream tasks such as knowledge base construction, question answering, document comprehension and so on [Liu and Croft 2002; Sen Wu et al. 2018; Geva and Berant 2018].

Most existing studies [Zhao, Wu, and Wang 2019; Lin et al. 2020; Yu et al. 2019] pre-define each *key* to be extracted as a class label, and then predict the class label of each word in the document. However, such standard protocol might be incapable to handle the keys that are omitted from the training set, since these *zero-shot keys* correspond to no training instances. Meanwhile, in real-world production, the amount of keys needed to extract might be indeed massive, and thus labeling large-scale training data for each key is both labor-intensive and unscalable [Li, Min, and Fu 2019]. In this scenario, we want to learn key-invariant feature representation such that the model can generalize to zero-shot keys without additional annotation cost.



GARDENIA BAKERIES (KL) SDN BHD (139386 X)
Lot 3, Jalan Pelabur 23/1,
40300 Shah Alam, Selangor.
Tel 03- 55423228 Fax:03- 55423213
GST ID: 000381399040

TAX INVOICE / ADJUSTMENT NOTE

Cash Inv No.: 7819F712
VE0514 Date: 19/08/2017
MAKASSAR FRESH MARKET SDN BHD
GROUND FLOOR, NO. 4 & 6,
JALAN SS 15/4B,
47500 SUBANG JAYA, SELANGOR
VE05: Ridzuan(11900)
DD: 19/08/2017 10:38

Description	U.P	Iss	Exc	D	Sale	Amt(RM)
O.C. WHITE	2.13	5	3	0	2	4.26
WHOLEMEAL	2.78	3	0	0	3	8.34
O.C JUMBO	2.97	5	0	0	5	14.85
Total 0% supplies:						27.45
DELICIA-B/SCOTCH	3.72	2	0	0	2	7.44
CR-B'SCOTCH	0.72	20	0	0	20	14.40
Total 6% supplies (excl. GST):						21.84
GST:						1.31
Total 6% supplies (Inc. GST):						23.15
Total 0% supplies:						27.45
Total Payable						50.60

E.&.O.E.
Received above goods in good order condition.
The recipient of Gardenia's products is required to make necessary adjustments to its input tax claims, on the basic of the adjustments shown in this Tax Invoice / Adjustment Note
CUSTOMER'S COPY

Gerbang Alaf Restaurants Sdn Bhd
(65351-M)
Licensee of McDonald's
Level 6, Bangunan TH, Damansara Uptown3
No.3, Jalan SS21/39,47400 Petaling Jaya
Selangor
(GST ID No: 000504664064)
McDonald's Rawang DT(#188)
Tel No. 03-6092-1968
TAX INVOICE

MFY Side 1

INV# 001881904580783
ORD #70 -REG #19- 18/03/2018 23:02:16

QTY ITEM	TOTAL
1 L McChicken	11.10
1 L Coke	
1 L Fries	
2 Small Cone	2.00
TakeOut Total (incl GST)	13.10
Total Rounded	13.10
Cash Tendered	20.00
Change	6.90
TOTAL INCLUDES 6% GST	0.74

nk You and Please Come Again.
Guest Relations Center : 1300-13-1300

Figure 1: Examples of two labeled receipt documents. The green box, blue box and red box represent key, trigger and value, respectively.

On the other hand, when a human searches for the value of a given key from a document, she always recognizes certain words or phrases that act as cues in advance. For instance, "total payable" and "total rounded" are distinct cue phrases of the same key "total" (shown in Figure 1). Then, the values "50.60" and "13.10" are easy to extract since they locate in the same row as the cue phrase according to the implicit tabular format. For clarity, we call such cue phrase as *trigger*, which can be defined as continuous or discontinuous words in the document that explain the key suitably and instruct the proper position of the value. Note that, the trigger should be a sufficient evidence to recognize the value even if the value is replaced with some random words. Previous studies often leverage the attention mechanism for more ac-

]

curate extraction. To explore whether the learned features in previous models focus on the trigger, we mask the trigger and let the models predict again. However, the results in Section show that these models can still extract the correct value. That is to say, the previous models do not depend on the trigger to recognize the value, thus cannot reflect the *true* proxy of explanations on why humans recognize the value.

To address these issues, we propose a novel two-stage architecture, named Key-Aware and Trigger-Aware (KATA), for this task. Since no labeled instances to the zero-shot keys are available, some auxiliary information is needed to represent these keys [Wang et al. 2019]. We assume that the words contained in each given key give a concise and precise description for the content to extract. Then, we treat each key not simply as an atomic label, but instead represent it as *a sequence of words* such that the semantic and visual relationships between the given key and the words in the document can be explicitly learned. For example, for the left receipt in Figure 1 with the semantics in the given key "telephone" it might be easier to recognize its true value "03-55423228". Then, with the key as the input, this problem is transformed as a 0/1 classification task over the words in a document, resulting in that the training data with various keys from different domains can be collaboratively learned.

Furthermore, inspired by how humans recognize the value, the KATA model explicitly learns two mappings, namely from key representations to trigger representations and then from trigger representations to target values. Specifically, KATA extracts the trigger explicitly in the first stage (key-to-trigger mapping) and recognizes the value based on the predicted trigger in the second stage (trigger-to-value mapping). Note that some keys may not have the corresponding triggers in the document. For example, in the right receipt in Figure 1 the value "18/03/2018" to the key "date" does not have trigger in the document since it has provided sufficient discriminative information. Thus, we also add key as the auxiliary input in the second stage to let the model learn the direct mapping from key representations to values. The reason why KATA could accurately recognize zero-shot keys might be that these learned mappings are intrinsic and invariant across different keys and documents.

Although we propose the KATA model to address the aforementioned issues, then another question is how to annotate triggers and values for large-scale documents to train this model. Wikipedia, the largest online encyclopedia to date, is widely used for distant supervision and model pre-training in many existing studies [Nguyen and Moschitti 2011; Chen et al. 2017]. We discover that *Wikipedia Infobox*, frequently used to list some facts as a table of attribute-value pairs [Lehmann et al. 2015], can be tailored in this work. Thus, we automatically construct Wikipedia Infobox datasets to pre-train these two mentioned mappings.

Based on two target datasets - the SROIE dataset containing 972 English documents and the Grater dataset containing 4,032 Chinese documents, we compare the pre-trained KATA model and other baseline models. The empirical experiment demonstrates that KATA achieves the best 0.7069 and 0.7339 accuracy for extracting zero-shot keys in the SROIE and Grater dataset, respectively, while all the previous methods get close to 0% accuracy. Moreover, KATA also obtains accuracy improvement on non-zero-shot keys compared with baseline models. We also demonstrate that removing index position embeddings in the model backbone and using parameters with more pre-training epochs for initialization will improve the accuracy of extraction.

Our contributions in this paper are as follows:

- To the best of our knowledge, this paper is the first work to extract the values of both zero-shot and non-zero-shot keys in form-like documents.
- We propose a novel two-stage architecture to employ the semantics of the given keys and predict triggers explicitly.
- We construct the large-scale labeled datasets from Wikipedia Infobox to pre-train the model and fine-tune it on two target datasets. The experimental results show that the proposed KATA model obtains great improvement in accuracy compared with baseline models.

Key-Aware and Trigger-Aware Information Extraction

In this section, we introduce basic concepts and their notations in Section . Then we explain why and how we make key aware and trigger aware in Section and Section . Finally, we present specific modules in KATA in Section and how to pre-train KATA in Section .

Problem Formulation

We first denote $\mathcal{K}_n = \{k_i\}_{i=1}^{N_n}$ and $\mathcal{K}_z = \{k_i\}_{i=1}^{N_z}$ as the set of non-zero-shot keys and zero-shot keys, respectively. Note that $\mathcal{K}_n \cap \mathcal{K}_z = \emptyset$. Then, we define the labeled training set as a collection of 4-tuples, $\mathcal{D}_{tr} = \{(k_i, d_i, t_i, v_i) | k_i \in \mathcal{K}_n\}_{i=1}^{N_{tr}}$ and the unlabeled test set as a collection of 2-tuples, $\mathcal{D}_{te} = \{(k_i, d_i) | k_i \in \mathcal{K}_n \cup \mathcal{K}_z\}_{i=1}^{N_{te}}$. Here k_i, d_i, t_i, v_i denote key, document, trigger, and value, respectively. The question now is to learn a model that is trained with the training set $\mathcal{D}_{tr}$, but can still recognizes the value of key in the test set $\mathcal{D}_{te}$, regardless of the key is zero-shot or non-zero-shot.

Key-Aware Extraction

Most previous studies [Zhao, Wu, and Wang 2019; Lin et al. 2020; Yu et al. 2019] pre-define a set of keys by considering them only as the atomic class. Taking LayoutLM [Xu et al. 2020a] as an example, it creates text embedding and 2-D position embedding of each word in the document, where 2-D position embedding is used to model the relative spatial position. Then, LayoutLM employs a multi-layers Transformer [Vaswani et al. 2017] to capture features and uses SoftMax to classify each word into a class label. However, such classic solution fails to predict the keys that are omitted from the training set, since these zero-shot keys correspond to no training instances.

To enable the model to predict these zero-shot keys, some auxiliary information is needed to represent them [Wang et al. 2019]. To this end, instead of treating each key as simply an atomic label, we characterize them with the semantic of each key, which is defined as a sequence of words that can be regarded as a concise but precise description for the

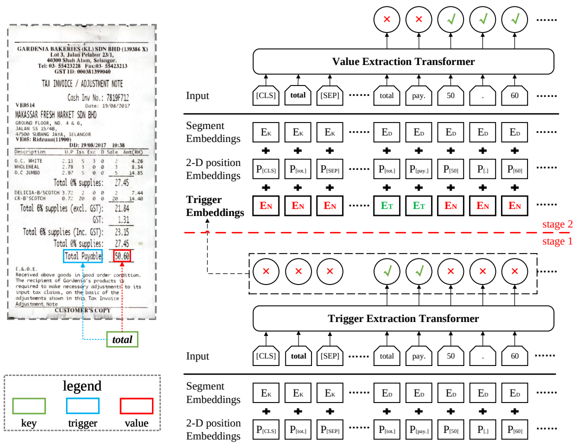


Figure 2: Architecture of the proposed KATA model. The left part presents an example of receipt and the right part presents the trigger extraction module and value extraction module in KATA based on this example.

content to extract. For example, in Figure 2 to extract the information about "total consumption in this market", we use string "total" to represent the name of the key. Then, the model predicts "50.60" as the value of "total", which can be transformed as a 0/1 classification task over the words in the document. In other words, the input key informs the model what information to extract and the model can recognize the value via analyzing the semantic and visual relationship between the given key and the words in the document.

Trigger-Aware Extraction

When a human searches for the value of a given key from a document, she always recognizes some continuous or discontinuous words that act as cues in advance. We call such cue phrase as trigger, which explains the key suitably and instructs the proper position of the value. For example, in Figure 2 to extract the value of "total", the phrase "total payable" acts as the trigger since it is the most proper phrase in the document to explain "total".

Then, we aim to explore whether the previous models focus on the trigger. Following [Zeiler and Fergus 2014], for a key k and a document d, we mask the trigger t of k and let the model predict the value again. Here the content of the masked words in trigger is replaced with "[UNKNOWN]" and the position of the words is retained. Figure 4 shows the results before and after masking triggers of the key "total". LayoutLM only obtains 0.0457 F1 decrease, which means that LayoutLM does not depend on the trigger, but might depend on integral layout features to recognize the value.

To make the model focus on the trigger, the proposed KATA model aims to explicitly extract trigger and recognizes the value based on the predicted trigger. Recall the above example, "total payable" acts as the trigger of the key "total" since they have similar semantic representation. Then, KATA recognizes "50.60" as the value of "total" since they locate in the same row according to implicit tabular format. At a high-level, KATA learns two mappings, namely key-to-trigger and trigger-to-value. Learning key-to-trigger mapping only relies on the semantic relationship between the key and trigger and irrespective of where the trigger locates on the page. Learning trigger-to-value mapping depends on both the semantic relationship and the relative, yet not absolute, position relationship between trigger and value. Thus, these two mappings might be intrinsic and invariant across different keys and documents in general.

Formally, we can expand the probability $P(v|d,k)$ through the total probability formula as follows,

$$P(v|d,k) = \sum_{t} P(t|d,k) \cdot P(v|d,k,t), \qquad (1)$$

where, $P(t|d,k)$ and $P(v|d,k,t)$ represent key-to-trigger mapping and trigger-to-value mapping, respectively. Afterwards, we use two attention-based modules to model these two probabilities or mappings.

Trigger Extraction Stage and Value Extraction Stage

Figure 2 depicts the architecture of the proposed KATA model with an example of a receipt document. KATA is composed of two stages, namely trigger extraction stage and value extraction stage.

The trigger extraction stage receives a given key k and a document d, and then extracts the trigger t from the words in this document. This process can be regarded as binary classification ("Trigger" or "None") over the words in the document. Following BERT framework [Devlin et al. 2019], we pack the words in the key and the words in the document together, then tokenize them into one token sequence. To make the model differentiate the key and document, we add a token "[SEP]" between them and assign a learned segment embedding to each token indicating whether it belongs to the key or document. To represent the spatial position of each token, we assign 2-D position embedding to each token following LayoutLM [Xu et al. 2020a].

The value extraction stage receives a key k, a document d, the predicted trigger t, and then recognizes the value v from the words in this document. This process can also be regarded as binary classification ("Value" or "None") over the words in the document. Note that the key is added as auxiliary information to the value extraction stage once extracting incorrect triggers or no trigger exists in the previous stage. Since the trigger is predicted explicitly in the first stage, we allocate learned trigger embedding to each token indicating whether it belongs to a trigger or not.

In both trigger extraction and value extraction stage, the input representation of each token is constructed by summing all the corresponding embeddings. Then, a multilayers Transformer [Vaswani et al. 2017] is used to extract the interaction features among tokens. The structure of the Transformer is the same but the parameters are different in two stages. Finally, we use SoftMax to classify each token and use cross-entropy loss as the objective function.

Pre-training KATA based on Wikipedia

In this section, we use Wikipedia, the largest online encyclopedia to date, to pre-train two mentioned mappings for the

KATA model. Wikipedia Infobox is frequently used to list an article's most relevant facts as a table of attribute-value pairs on Wikipedia page [Lehmann et al. 2015]. To obtain content and position information of each word in Wikipedia Infobox, we extract the Infobox section from the original Wikipedia page, re-render this section to a PDF document via *wkhtmltopdf*[1], and then apply *PDFMiner*[2] to parse the PDF document.

To label the triggers and values on corresponding words, we employ the RDF statements data from DBpedia [Lehmann et al. 2015]. Here, RDF statements consist of a large number of RDF triplets (d, c, v), where d, c and v represent document id, ontology class and value, respectively. In detail, DBpedia develops an *ontology* concept, which consists of thousands of hierarchical classes, and maps Wikipedia Infobox attributes to one of the class labels by community effort. Based on this data, we assume that an ontology class c and a key k are mutually exchangeable. Then, we assign a label "Value" to the words in the documents that have exact string or link matching with v. Then, we assume that the trigger and its value always locate in the same row of the Infobox table. Thus, we only keep those Infobox tables with only two columns. For each row in the Infobox table, if one value locates at the second cell, we set the words in the first cell as its corresponding trigger. Finally, we can obtain the content, position and annotation information of each word in the document.

Based on the above process, we automatically construct the Wikipedia Infobox dataset to pre-train the trigger extraction module and value extraction module separately. Note that we use labeled triggers as the input to train the value extraction module to prevent the prediction errors in the trigger extraction stage. Here, we emphasize that the trigger-value pairs in this pre-trained data always have left-to-right structure, which has limitations to handle the pairs with top-bottom or other complex structures. Although the Wikipedia Infobox dataset is enough to handle two target datasets in this paper since most trigger-value pairs in these two datasets are left-to-right, we aim to add more pre-trained pairs with abundant layout structure, such as from *Wikitable*, to serve a broader situation in the future.

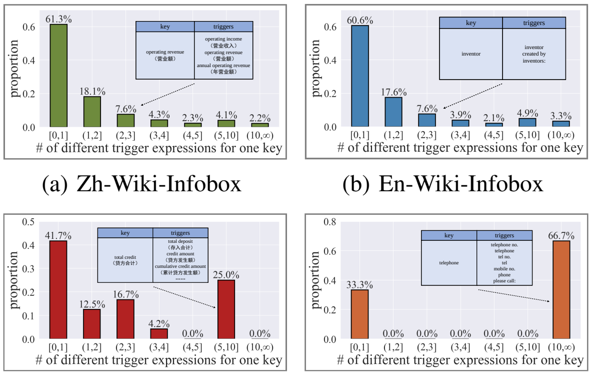
Experiments

Datasets

In this paper, we use two target KIE datasets, a publicly accessible *SROIE* dataset[3] and a private *Grater* dataset, for fine-tuning and evaluation. The documents are English receipts and invoices in the SROIE dataset and Chinese bank statements in the Grater dataset. Therefore, we construct two Wiki-Infobox datasets in English and Chinese, called *En-Wiki-Infobox* and *Zh-Wiki-Infobox*, for pre-training KATA. The number of keys, documents, and key-value pairs of each dataset is listed in Table 1. The Annotation of triggers in the SROIE and Grater dataset does not incur significant additional effort because the triggers are typically short phrases.

dataset	#key	#document	#key-value pair
Zh-Wiki-Infobox	6,151	281,281	897,520
En-Wiki-Infobox	7,370	398,467	1,238,281
Grater	24	4,032	18,825
SROIE	6	972	5,505

Table 1: The statistic of different datasets.



(a) Zh-Wiki-Infobox

(b) En-Wiki-Infobox

(c) Grater

(d) SROIE

Figure 3: The distribution on the number of different trigger expressions for each key in different dataset. The blue table shows the example of the correspondence of key and trigger.

Recall the examples in Figure 1, "total payable" and "total rounded" are distinct *trigger expressions* for the same key "total". Hence, we count the number of different trigger expressions for each key and draw its distribution for four datasets in Figure 3. We observe that around 40% keys correspond to more than one trigger expressions in two Wiki-Infobox datasets, and this ratio increases to around 60% and 70% in the SROIE and Grater dataset, respectively. This phenomenon shows that key-to-trigger mapping is one-to-many mapping. For each key, the mapping will be more difficult to learn along with the number of trigger expressions increases, according to the results in Section .

In consideration of the zero-shot learning, we select several keys as zero-shot keys and remove their corresponding key-value pairs from the training set but retain the pairs in the test set. To make the difficulty of the selected keys average, we select some keys with small number of trigger expressions and some keys with large number of trigger expressions. Note that the SROIE dataset includes 2 zero-shot keys (listed in Table 3) and 4 non-zero-shot keys ("company", "address", "date" and "total"). The Grater dataset includes 8 zero-shot keys (listed in Table 3) and 16 non-zero-shot keys (e.g. "account", "total credit", "currency", etc.).

Following [Xu et al. 2020b], the evaluation metric is the exact match of the value recognition results in both SROIE and Grater datasets. That is to say, value is predicted correctly when all the words in this value are exactly the same as the ground-truth.

Baselines and Proposed Models

- **BERT** [Devlin et al. 2019]. BERT model classifies each word in the document as one of the pre-defined keys or

[1] https://wkhtmltopdf.org/

[2] https://pypi.org/project/pdfminer/

[3] https://rrc.cvc.uab.es/?ch=13

row	method	the SROIE dataset						the Grater dataset					
		zero-shot keys			non-zero-shot keys			zero-shot keys			non-zero-shot keys		
		P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$
1	BERT [Devlin et al. 2019]	0	0	0	0.9099	0.9099	0.9099	0	0	0	0.9986	0.9965	0.9976
2	LayoutLM [Xu et al. 2020b]	0	0	0	0.9460	0.9460	0.9460	0	0	0	1.0000	1.0000	1.0000
3	KA-bert	0.1043	0.1000	0.1021	0.9373	0.9373	0.9373	0.2431	0.2422	0.2426	0.9993	0.9979	0.9986
4	KA-wiki	0.6468	0.5961	0.6204	0.9469	0.9430	0.9449	0.3768	0.3702	0.3735	1.0000	1.0000	1.0000
5	KATA-bert	0.0649	0.0471	0.0545	0.9366	0.9382	0.9347	0.3692	0.2491	0.2975	0.9993	0.9993	0.9993
6	KATA-wiki	**0.7678**	**0.6549**	**0.7069**	**0.9483**	**0.9457**	**0.9470**	**0.7378**	**0.7301**	**0.7339**	1.0000	1.0000	1.0000
7	KATA-wiki w/ index embedding	0.6569	0.6000	0.6283	0.9455	0.9423	0.9439	0.6809	0.6644	0.6725	1.0000	1.0000	1.0000
8	KATA-wiki w/o fine-tuning	0.5116	0.4314	0.4681	0.1106	0.0502	0.0691	0.1322	0.0796	0.0994	0.0963	0.0594	0.0735

Table 2: Comparing KATA with baseline models.

data	zero-shot keys	#trigger	trigger-F1	value-F1
SROIE	cash	36	0.5689	0.6730
	telephone	17	0.8537	0.7500
Grater	organization	5	0.7059	0.6525
	balance in previous page	3	0.6538	0.8269
	subject	2	0.4783	0.8696
	card number	2	0.4375	0.9375
	balance	1	1.0000	0.8750
	unit	1	0.9804	0.5743
	print method	1	1.0000	1.0000
	credit count	1	1.0000	1.0000

Table 3: Results of different zero-shot keys in KATA.

“None”. Note that BERT cannot work for zero-shot keys since there are no training samples for these keys.

- **LayoutLM** [Devlin et al. 2019]. Compared with the BERT model, the LayoutLM model adds 2-D position embeddings to each word and is pre-trained on a large-scale dataset. LayoutLM also considers this problem as multi-classification and cannot work for zero-shot keys.
- **KA**. Our proposed key-aware extraction model (KA) receives the given key as input and then directly classifies each word in the document as “Value” or “None”. There are two variants, called KA-bert and KA-wiki. KA-bert uses the pre-trained BERT parameters for initialization and KA-wiki uses the parameters pre-trained on En-Wiki-Infobox for fine-tuning the SROIE dataset and Zh-Wiki-Infobox for fine-tuning the Grater dataset.
- **KATA**. Our proposed KATA model is a two-stage model. With the key as input, KATA first extracts the trigger and then recognizes the value based on the predicted trigger. There are also two variants, called KATA-bert and KATA-wiki, which are similar to KA above.

Experimental Results

In this section, we first compare the overall performance of the baseline models and the KATA model. Then, we do ablation studies to evaluate the influence of each module in KATA. Finally, we compare the performance of the pre-trained parameters with different training epochs in KATA.

To compare the accuracy for zero-shot keys of baseline models and KATA, we present the results in row 1, 2 and 6 of Table 2. KATA-wiki model obtains 0.7069 F1 value in the SROIE dataset and 0.7339 F1 value in the Grater dataset for zero-shot keys. However, previous models fail to predict zero-shot keys in both datasets since there are no training samples for these keys. In detail, Table 3 demonstrates the trigger-F1 and value-F1 for each zero-shot key. Overall, the keys with less number of trigger expressions obtain higher accuracy since low diversity of trigger expressions means less difficulty. Even though, for some zero-shot keys that correspond to tens of trigger expressions in the SROIE dataset, KATA also obtains more than 0.6 F1 value. Note that, there is a special case “unit”. As shown in Figure 5, the predicted value of key “unit” includes redundant information sometimes and we further analyze it in Section . Overall, the KATA-wiki model has capability to recognize the value of zero-shot keys.

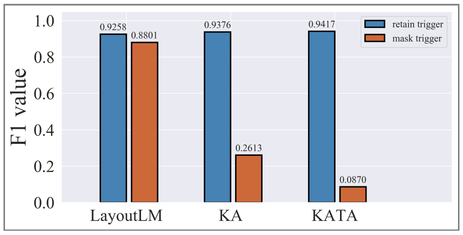


Figure 4: Comparing the F1 value of key “total” for retaining and masking the trigger of “total” in different models.

Meanwhile, we compare the accuracy for non-zero-shot keys of baseline models and KATA. In the SROIE dataset, BERT, LayoutLM, and KATA-wiki model obtain 0.9099, 0.9460, and 0.9470 F1 value respectively for non-zero-shot keys. In the Grater dataset, BERT, LayoutLM, and KATA-wiki model obtain 0.9976, 1.0000, and 1.0000 F1 value respectively for non-zero-shot keys. That is to say, the KATA-wiki model also obtains comparable performance on non-zero-shot keys with other baselines models. Note that, most models obtain close to 1.0 F1 value for non-zero-shot keys in the Grater dataset. The reason is that the model is easy to fit based on some training samples, due to the simple document layout structure. To sum up, compared with other baseline models, the proposed KATA-wiki model obtains great improvement in accuracy for zero-shot keys and comparable performance for non-zero-shot keys.

To explore the influence of pre-trained parameters, we

row	method	the SROIE dataset						the Grater dataset					
		zero-shot keys			non-zero-shot keys			zero-shot keys			non-zero-shot keys		
		P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$
1	KATA-wiki (10k batches)	0.3224	0.2510	0.2822	0.9394	0.9355	0.9375	0.4948	0.4948	0.4948	1.0000	1.0000	1.0000
2	KATA-wiki (20k batches)	0.4082	0.3314	0.3658	0.9483	**0.9457**	**0.9470**	0.5261	0.5225	0.5243	1.0000	1.0000	1.0000
3	KATA-wiki (30k batches)	0.6522	0.5588	0.6019	0.9467	0.9396	0.9431	0.5725	0.5467	0.5593	0.9993	0.9993	0.9993
4	KATA-wiki (1 epoch)	0.5952	0.5882	0.5917	0.9460	0.9396	0.9428	0.6125	0.6125	0.6125	1.0000	1.0000	1.0000
5	KATA-wiki (2 epochs)	0.7028	0.6490	0.6748	**0.9488**	0.9437	0.9463	0.6401	0.6401	0.6401	1.0000	1.0000	1.0000
6	KATA-wiki (3 epochs)	**0.7678**	**0.6549**	**0.7067**	0.9429	0.9430	0.9416	**0.7378**	**0.7301**	**0.7339**	1.0000	1.0000	1.0000

Table 4: Comparing different epochs of pre-trained parameters in KATA.

present the results in row 3, 4 and row 5, 6 of Table 2. For KA, using pre-trained Wiki-Infobox parameters for initialization obtains 0.5183 and 0.0076 F1 improvement for zero-shot keys and non-zero-shot keys in the SROIE dataset, 0.1309 and 0.0014 F1 improvement for zero-shot keys and non-zero-shot keys in the Grater dataset. For KATA, using pre-trained Wiki-Infobox parameters for initialization obtains 0.6524 and 0.0123 F1 improvement for zero-shot keys and non-zero-shot keys in the SROIE dataset, 0.4364 and 0.0007 F1 improvement for zero-shot keys and non-zero-shot keys in the Grater dataset. That is to say, for both KA and KATA model, compared with using BERT parameters, using pre-trained Wiki-Infobox parameters for initialization obtains great improvement on zero-shot keys and also obtains improvement on non-zero-shot keys.

KATA explicitly predicts trigger to mimic how humans recognize the value. To explore the influence of this process, we present the results in row 4 and 6 of Table 2. Compared with KA-bert, KATA-wiki obtains 0.0865 and 0.3604 F1 improvement for zero-shot keys in the SROIE dataset and Grater dataset, and 0.0021 F1 improvement for non-zero-shot keys in the SROIE dataset. That is to say, explicitly predicting trigger obtains great improvement for zero-shot keys nd also obtains improvement on non-zero-shot keys.

Furthermore, to explore whether the proposed model focuses on trigger to recognize value, we mask the trigger and then let the model predict the value again (as mentioned in Section). Figure 4 shows the results before and after masking triggers of the key "total" in different models. LayoutLM only obtains 0.0457 F1 decrease, which means that LayoutLM does not depend on the trigger to recognize the value. With the key as input, KA obtains 0.6763 F1 decrease, which means that KA pays more attention to the triggers. By explicitly predicting triggers, KATA obtains 0.8547 F1 decrease. That is to say, the learned attention in KATA reflects the true proxy of explanations on how humans would recognize the value via the trigger.

LayoutLM packs the words in the document as an ordered sequence and assigns a learned index embedding to characterize the position of each word. Conventional top-bottom left-to-right order is used to represent the order of words [Meunier 2005], however, cannot always agree with the true reading order of the document. Thus, to explore the influence of incorrect index embedding, we present the results in row 6 and 7 of Table 2. Compared with KATA-wiki, adding index position embeddings obtains 0.0786 and 0.0614 F1 decrease for zero-shot keys in the SROIE dataset and Grater dataset, and 0.0031 F1 decrease for non-zero-shot keys in the SROIE dataset. That is to say, removing index embedding obtains improvement in accuracy for both zero-shot keys and non-zero-shot keys. Therefore, we remove index embedding of each token in the KATA model.

We directly evaluate the results based on pre-trained Wiki-Infobox parameters without fine-tuning and present the results in row 6 and 8 of Table 2. Apparently, testing without fine-tuning obtains quite bad accuracy. It demonstrates that the documents in two Wiki-Infobox datasets have a different distribution with the documents in two KIE datasets. Nevertheless, using pre-trained Wiki-Infobox parameters for initialization still obtains great improvement in accuracy for both zero-shot and non-zero-shot keys.

To compare the accuracy of pre-trained Wiki-Infobox parameters with different training epochs, we compare 6 pre-trained parameters and each of them is 10,000 batches, 20,000 batches, 30,000 batches, 1 epoch, 2 epochs, and 3 epochs (as shown in Table 4). In general, the overall accuracy is monotonically increased as more batches and epochs are trained during the pre-training step. Therefore, we use parameters with more pre-training epochs for initialization in the KATA model.

Case Study and Limitations

We give some examples of correct-predicted and incorrect-predicted results when predicting zero-shot key "cash" from the SROIE dataset in Figure 5. For SROIE A and SROIE B page, KATA correctly predicts the triggers and values. Note that, although the trigger in SROIE B page is "payment", which is the synonyms of key "cash", KATA can still predict correctly. However, for SROIE C page, KATA finds out incorrect trigger-value pair "cash-receipt" and omits the true pair "MASTER-165.00". One reason for this limitation might be that we do not explicitly limit the type of value, thus KATA predicts "receipt" as value by mistake. Another reason might be that KATA only receives one type of key expression as input, which neglects "MASTER" as the true trigger. For Grater A page, although KATA extracts the correct trigger of key "unit", it includes redundant information "date". The reason might be that KATA never sees "unit" in pre-training and training data, thus it has no knowledge about the feature of the value. Thus KATA may guess that the value will appear following the trigger "unit". However, the true value "RMB" has close distance with other redundant information, thus the predicted value includes this redundant information. If the model limits the type of value, it

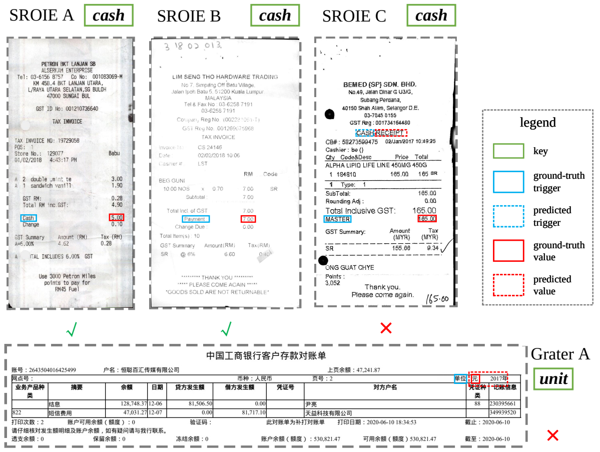

SROIE A
cash
SROIE B
cash
SROIE C
cash
legend
key
ground-truth trigger
predicted trigger
ground-truth value
predicted value
√
√
×
Grater A
unit
×

Figure 5: Examples of different documents. The green box represents key. The blue-solid, blue-dashed, red-solid and red-dashed boxes represent the ground-truth trigger, predicted trigger, ground-truth value and predicted value.

will predict correctly. Therefore, we have two ideas to tackle these limitations. On one hand, inspired with [Majumder et al. 2020], we can filter some wrong values by limiting the type of the target value. On the other hand, we except the model can integrate several distinct key expressions as input to adapt different trigger expressions for one key.

Experimental Conclusion

In conclusion, compared with baseline models, KATA obtains great improvement in accuracy for zero-shot keys and comparable accuracy for non-zero-shot keys. To obtain the best performance of KATA, we use pre-trained 3-epochs Wiki-Infobox parameters for initialization, explicitly predict trigger, and remove index position embeddings.

Related Work

Key Information Extraction

Recently, significant studies have focused on information extraction from unstructured or semi-structured data, in the form of plain texts, digital documents, and web pages [Nanyun et al. 2017; Ferrara et al. 2014; Palm, Winther, and Laws 2017]. Plain texts mainly refer to natural language sentences, which only contain textual information, while digital documents and web pages contain both textual and visual information.

In this paper we focus on information extraction from digital documents or web pages. Early works usually require a great number of human-crafted rules or patterns, which is only used for particular documents and difficult to generalize [Medvet, Bartoli, and Davanzo 2011; Gulhane et al. 2011; Gentile, Zhang, and Ciravegna 2013]. To eliminate rigorous rules and patterns, recent studies deploy learning-based methods via various popular techniques such as CNNs, GCNs, and Transformers. Katti et. al. [Katti et al. 2018] and Zhao et. al. [Zhao, Wu, and Wang 2019] employ CNNs to extract key information from PDF documents. Liu et. al. [Liu et al. 2019] and Yu et. al. [Yu et al. 2020] apply GCN to extract features upon the topological of every word, and use Bi-LSTM and CRF to classify each word an information category. The difference is that Yu's model uses another CNN to extract visual features on the document image, and concatenate this feature with the GCN feature. Majumder et. al. [Majumder et al. 2020] use attention-based network to extract textual and visual features of neighbour words and classify the key category of the target. Lin et. al. [Lin et al. 2020] focus on structured information extraction on web documents. They use CNN and Bi-LSTM to predict each word a class label in the first stage, and then another neural network is used to captures longer range distance and semantic relatedness between the information extracted in the previous stage. Large-scale pre-training models, like BERT, become the state-of-the-art techniques on challenging NLP tasks. To employ this idea on information extraction in digital documents, Xu et. al. [Xu et al. 2020b] propose LayoutLM to jointly model the interaction between text and layout information across document images. LayoutLM achieves great improvements in the KIE tasks.

Zero Shot Learning

Zero-shot learning is a promising learning paradigm proposed by Palatucci et. al. [Palatucci et al. 2009], where the possible values for the class Y include values that have been omitted from the training examples. Recently, extensive works aim to tackle zero-shot learning in various tasks, such as relation extraction, entity extraction, image recognition, etc. [Palatucci et al. 2009; Pasupat and Liang 2014; Levy et al. 2017; Xian et al. 2018; Lockard et al. 2020].

Pasupat et. al. [Pasupat and Liang 2014] consider a zero-shot task of extracting entities from web pages. The traditional entity extraction task requires information such as seed entities and then extracts the target entities that are similar to these seeds. Differently, Pasupat's model uses a natural language query to replace these seed entities, and classifies each candidate words as “entity” or “None”. Levy et. al. [Levy et al. 2017] aim to extract relation in natural language sentences that are only specified at test-time. To this end, they use question templates to replace the given knowledge-base relation and use distant supervision for a relatively large number of relations from Wikipedia. Lockard et. al. [Lockard et al. 2020] propose a solution for zero-shot open-domain relation extraction from web pages in the unseen website and unseen vertical. To generalize to these never-before-seen templates and topics, they propose a graph neural network model that encodes semantic textual and visual patterns across different training websites.

Conclusion

In this paper, we focus on extracting the values of zero-shot keys in form-like documents. We propose a novel extraction model, named KATA. With the input key, it explicitly learns key-to-trigger and trigger-to-value mappings. We also automatically construct two datasets for pre-training two mappings. The experiments on two target KIE datasets demonstrate the effectiveness of KATA on extracting zero-shot keys.

Acknowledgements

The research work supported by the National Key Research and Development Program of China under Grant No. 2017YFB1002104, the National Natural Science Foundation of China under Grant No. 62076231, U1811461. We thank Xu Wang and Jie Luo (from P.A.I Tech) for their kind help at the beginning of the project, and Ganbin Zhou and Fen Lin (from WeChat Search Application Department, Tencent) for their useful advices for the paper. We also thank anonymous reviewers for their valuable comments and suggestions.

References

Chen, D.; Fisch, A.; Weston, J.; and Bordes, A. 2017. Reading wikipedia to answer open-domain questions. In *ACL*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Ferrara, E.; De Meo, P.; Fiumara, G.; and Baumgartner, R. 2014. Web data extraction, applications and techniques: A survey. *Knowledge-based systems* .

Gentile, A. L.; Zhang, Z.; and Ciravegna, F. 2013. Web Scale Information Extraction with LODIE. In *NCAI*.

Geva, M.; and Berant, J. 2018. Learning to Search in Long Documents Using Document Structure. In *ACL*.

Gulhane, P.; Madaan, A.; Mehta, R. R.; Ramamirtham, J.; Rastogi, R.; Satpal, S.; Sengamedu, S. H.; Tengli, A.; and Tiwari, C. 2011. Web-scale information extraction with vertex. In *ICDE*.

Huang, Z.; Chen, K.; He, J.; Bai, X.; Karatzas, D.; Lu, S.; and Jawahar, C. 2019. ICDAR 2019 competition on scanned receipt ocr and information extraction. In *ICDAR*.

Katti, A. R.; Reisswig, C.; Guder, C.; Brarda, S.; Bickel, S.; Höhne, J.; and Faddoul, J. B. 2018. Chargrid: Towards understanding 2d documents. In *EMNLP*.

Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P. N.; Hellmann, S.; Morsey, M.; Van Kleef, P.; Auer, S.; et al. 2015. DBpedia - a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic web* .

Levy, O.; Seo, M.; Choi, E.; and Zettlemoyer, L. 2017. Zero-shot relation extraction via reading comprehension. In *CoNLL*.

Li, K.; Min, M. R.; and Fu, Y. 2019. Rethinking zero-shot learning: A conditional visual classification perspective. In *ICCV*.

Lin, Y.; Ying, S.; Vo, N.; and Sandeep, T. 2020. FreeDOM: A Transferable Neural Architecture for Structured Information Extraction on Web Documents. In *KDD*.

Liu, X.; and Croft, W. B. 2002. Passage retrieval based on language models. In *CIKM*.

Liu, X.; Gao, F.; Zhang, Q.; and Zhao, H. 2019. Graph convolution for multimodal information extraction from visually rich documents. In *NAACL*.

Lockard, C.; Shiralkar, P.; Dong, X. L.; and Hajishirzi, H. 2020. ZeroShotCeres: Zero-Shot Relation Extraction from Semi-Structured Webpages. *ArXiv* .

Majumder, B. P.; Potti, N.; Tata, S.; Wendt, J. B.; Zhao, Q.; and Najork, M. 2020. Representation Learning for Information Extraction from Form-like Documents. In *ACL*.

Medvet, E.; Bartoli, A.; and Davanzo, G. 2011. A probabilistic approach to printed document understanding. *IJDAR* .

Meunier, J.-L. 2005. Optimized XY-cut for determining a page reading order. In *ICDAR*.

Nanyun, P.; Hoifung, P.; Chris, Q.; Kristina, T.; and tau Yih, W. 2017. Cross-sentence n-ary relation extraction with graph lstms. In *ACL*.

Nguyen, T.-V. T.; and Moschitti, A. 2011. End-to-end relation extraction using distant supervision from external semantic repositories. In *ACL*.

Palatucci, M.; Pomerleau, D.; Hinton, G. E.; and Mitchell, T. M. 2009. Zero-shot learning with semantic output codes. In *NeurIPS*.

Palm, R. B.; Winther, O.; and Laws, F. 2017. Cloudscan-a configuration-free invoice analysis system using recurrent neural networks. In *ICDAR*.

Pasupat, P.; and Liang, P. 2014. Zero-shot entity extraction from web pages. In *ACL*.

Sen Wu, L.; Hsiao, Xiao Cheng, B.; Hancock, T.; Rekatsinas, P.; Levis, C.; and Ré. 2018. Fonduer: Knowledge Base Construction from Richly Formatted Data. In *SIGMOD*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*.

Wang, W.; Zheng, V. W.; Yu, H.; and Miao, C. 2019. A Survey of Zero-Shot Learning : Settings , Methods , and Applications. *TIST* .

Xian, Y.; Lorenz, T.; Schiele, B.; and Akata, Z. 2018. Feature generating networks for zero-shot learning. In *CVPR*.

Xu, Y.; Li, M.; Cui, L.; Huang, S.; Wei, F.; and Zhou, M. 2020a. LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In *KDD*.

Xu, Y.; Li, M.; Cui, L.; Huang, S.; Wei, F.; and Zhou, M. 2020b. LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In *KDD*.

Yu, W.; Lu, N.; Qi, Xianbiao, G. P.; and Xiao, R. 2019. PICK: Processing Key Information Extraction from Documents using Improved Graph Learning-Convolutional Networks. *ArXiv* .

Yu, W.; Lu, N.; Qi, X.; Gong, P.; and Xiao, R. 2020. PICK: Processing Key Information Extraction from Documents using Improved Graph Learning-Convolutional Networks. In *ICPR*.

Zeiler, M. D.; and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *ECCV*.

Zhao, X.; Wu, Z.; and Wang, X. 2019. CUTIE: Learning to Understand Documents with Convolutional Universal Text Information Extractor. In *CVPR*.