Information extraction and document understanding
Extended abstract booklet

Martin Holeček

October 25, 2021

Introduction

The thesis is a compilation of two articles on the topic of information extraction - Holecek et al. [2019], Holeček [2021], and a previously unpublished technical report. The full source codes and an anonymized dataset Holecek [2020, 2019] are also important parts of this work.

This work has emerged from a favourable coincidence of multiple events:

1. Recently, all the new computational methods and improved hardware transformed the previously called times of "AI winter" into a "deep learning boom" Newquist [2018], Dargan et al. [2019].

2. The trending desire for further increase of automation levels in various corporate processes has set the information extraction task into a spotlight.

3. A presence of a novel, sufficiently big and cleaned dataset of annotated documents.

And so we can follow the path that the deep learning techniques have paved in different domains and tasks.

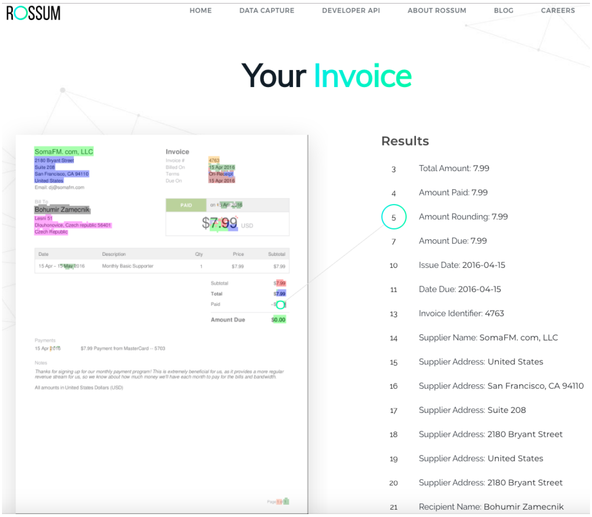
The information extraction task and the overall motivation A survey on information extraction methods Cowie and Lehnert

[1996] defines the task as: “Information Extraction starts with a collection of texts, then transforms them into information that is more readily digested and analyzed. It isolates relevant text fragments, extracts relevant information from the fragments, and then pieces together the targeted information in a coherent framework”.

The relevant collection of texts for this research are the texts in business documents such as invoices, pro forma invoices and debit notes. The targeted information is a classification of the texts that helps in automating various business processes – such as automated payment for invoices.

An example of a coherent framework's user interface can be seen in 1.

The exact numbers on the cost of automation of business documents processing are kept private. Existing approximations from some unofficial sources cos and Tenhunen and Penttinen [2010] give us an understanding, why the information extraction task is in the spotlight for many companies: It is reasoned, that even for a medium-sized company, the number of invoices processed per month approaches 25.000 and even a 1 % improvement in automatization leads up to 500$ monthly savings.



Your Invoice

SomaFM. com, LLC
2180 Bryant Street
Suite 208
San Francisco, CA 94110
United States
Email: dj@somafm.com

Invoice

Invoice #	4763
Billed On	15 Apr 2016
Terms	On-Receipt
Due On	15 Apr 2016

Bill To
Bohumir Zamecnik
Lesni 51
Dlouhonovice, Czech republic 56401
Czech Republic

PAID on 15 Apr 2016
$7.99 USD

Date	Description	Qty	Price	Subtotal
15 Apr – 15 May 2016	Monthly Basic Supporter	1	$7.99	$7.99

Subtotal	$7.99
Total	**$7.99**
Paid	-$
Amount Due	**$0.00**

Payments
15 Apr 2016 $7.99 Payment from MasterCard ··· 5703

Notes
Thanks for signing up for our monthly payment program! This is extremely beneficial for us, as it provides a more regular revenue stream for us, so we know about how much money we'll have each month to pay for the bills and bandwidth.

All amounts in United States Dollars (USD)

Results

3	Total Amount: 7.99
4	Amount Paid: 7.99
5	Amount Rounding: 7.99
7	Amount Due: 7.99
10	Issue Date: 2016-04-15
11	Date Due: 2016-04-15
13	Invoice Identifier: 4763
14	Supplier Name: SomaFM. com, LLC
15	Supplier Address: United States
16	Supplier Address: San Francisco, CA 94110
17	Supplier Address: Suite 208
18	Supplier Address: 2180 Bryant Street
19	Supplier Address: United States
20	Supplier Address: 2180 Bryant Street
21	Recipient Name: Bohumir Zamecnik

Figure 1: Example of an invoice and an extraction system together with its output. This example should also illustrate why invoices are called "structured documents". We can see that when the various information contained in the document is visually grouped, it usually belongs together. There is a heading "Invoice" under which segments of information about the invoice are written next to their explanations. Some rectangular areas do not have these explanations, and to find out what rectangular area speaks about the sender and supplier, one has to look for a small "Bill To:" heading. Please note, these specific rules apply only to this example and other invoices are notably different. (Online image source inv).

Table understanding in structured documents

We work with structured documents where not only the textual content but also the positioning of the texts does matter and no fixed set of layouts exists both in practice and in theory. Ultimately the documents feature a huge number of tables and nested table-like aligned structures.

Therefore to validate the hypothesis, that a parametrized system can learn and succeed at the goal of "extracting important information" we employ two goals:

- Detecting a specific so-called "line-item" table.
- And classification of other specified textual data like address, date, id details, amount types, tax details, ... (35 classes total).

Moreover, we desire a single efficient trainable end-to-end model, free of heuristic reasoning. In the research, we would show, that such a model can be successfully constructed.

We would validate all our assumptions about the needed layers and architecture modules, explore the importance of inputs and compare against a reasonable baseline. Note that since no referenced paper and/or commercial solution (d'Andecy et al. [2018], Coüasnon and Lemaitre [2014], Krieger et al. [2021], Riba et al. [2019], Lohani et al. [2018], Liu et al. [2019], Hamza et al. [2007]) can be easily customized to fit our aim successfully, we present our method as a novel approach and compare only against a logistic regression baseline. Other works focus on the exploration and usage of graph CNNs, which we will show is not enough in our case.

After successful validation of the initial hypothesis, the work would describe the ways to further improve the score by taking inspiration from other relevant concepts such as similarity and single-shot models Koch et al. [2015], Vinyals et al. [2016], query answer approaches Radford et al. [2019], Wang et al. [2017] and pairwise classification, or even dissimilarity Lin and Davis [2008]. Multiple deep learning models will be designed and evaluated and qualitative analysis will be carried.

More on the data and the annotated structured documents The annotation process is carried out by trained professionals with another supervisor and some automatic corrections. In the first part of the research, we use a smaller subset of the whole dataset (of 3554 PDFs) with annotated line-item tables to verify our assumptions. The whole bigger dataset (with 25071 PDFs) is used for simple validation of the training process and would be explored, as a whole, further in the final part.

All the documents are business documents like invoices, OCR'd documents are excluded to not measure a joint performance of OCR and any of our methods.

The splits are $\frac{3}{4}$ training and $\frac{1}{4}$ for validation. In the first part, an additional 83 annotated documents are used for testing with guaranteed different layouts (to measure true generalization).

Each document has around 500 words per page and 2 pages on average. To highlight that positioning of the words matters, they will be called "word-boxes" from now.

The metrics The metric used is inspired by Göbel et al. [2013] competition and selected to perform well with the following points:

The main unit of our focus is "word-boxes" (as opposed to char-boxes used in the competition). The problem is a multilabel-multiclass classification task (each word-box can be labelled with a line-item table class or other 37 classes) and the setting is as such that the labels and their positive occurrences are unbalanced (only 1.2 % positive occurrences of labels). Therefore the metric is chosen to be the F_1 score with 'micro' metrics aggregation rule (over more pages).

How to use the document's structural information Since no physical or mathematical model is available for reading documents, the network is present with all information possible:

- Geometrical:
  - Reading order of word-boxes (ordering of inputs; order for positional embedding).
  - Neighbouring 'seen' word-boxes (for graph convolution Riba et al. [2019]). See an example of a constructed graph over an invoice in 2.
  - Coordinates (for positional embedding, as in Vaswani et al. [2017]).
- Textual:
  - One-hot encoded characters (to allow the network to train its embeddings).
  - Features for capturing named entities (Chen et al. [2012], Nadeau and Sekine [2007], Abbasi et al. [2008]).
- Image features:
  - Whole image (for convolutional layers).
  - Crops around each word-box.

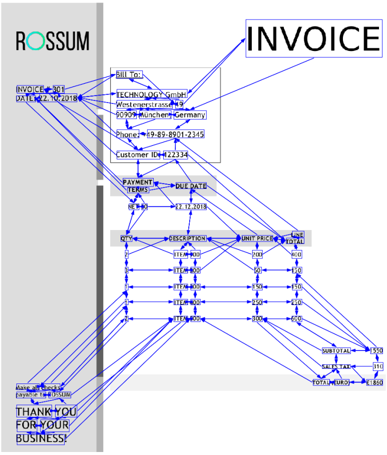


Figure 2: Sample invoice with edges defining neighbourhood word-boxes. Only the closest neighbour is connected for each word-box. (This invoice was artificially created for presentation and does not represent the invoices in the dataset.)

All the features for each word–box are then fed into the network (and can be perturbed during training for regularizations).

The model The model is depicted in 3, features blocks of graph CNN (over neighbours), CNN over sequence ordering and Multi-head attention. The final layers use sigmoidal activation function and binary cross-entropy as a loss.

Experimental results and conclusion of the first part A handful of experiments (ablation, input importance, comparison with a baseline) were designed and undertaken and the main results of the first part of the research are:

- Finding a line-items table is “easier” for the model than finding

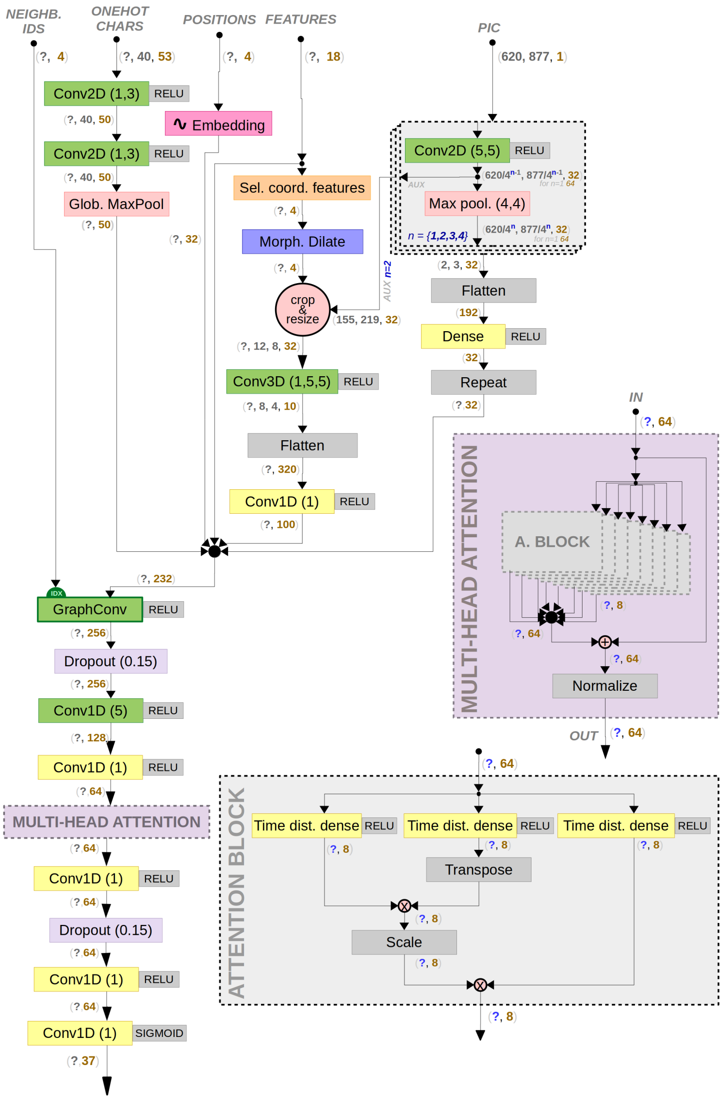


Figure 3: Simple data extraction model architecture for simultaneous table detection and all other information classification.

other structural information. Possibly due to class imbalance (holds even for a human).

- Logistic regression baseline improves with a higher number of neighbours but fails to generalize for “other” classes.
- Attention module is required for better generalization. All other parts of the architecture, as convolution over a sequence and graph CNN, are important for overall good results.
- The model performs well even on anonymized data and therefore: 1) the mutual position information is important and 2) the basic text features help the model generalize well.
- The model has been optimized on the smaller dataset, but it is verified that it can work equally well on bigger datasets.
- The tasks of finding line-items and other structural information do boost each other.

A brief exploration of generative models

To validate the value of the dataset annotation process and test the model from the first part on slightly different problems, a brief technical report on this topic is featured in the thesis. This part explores the problem from a different perspective - using a generative model based on expert knowledge (as opposed to a fully trainable classifier).

The expert knowledge gathered can be summarized in the following points:

- The interesting information to extract is usually located:
  - Near the explaining text that implicitly or explicitly defines the class (e.g. "IBAN:") or
  - In the same column that defines its class (e.g. header "amount") or
  - In the same usual place on a page (e.g. page numbers)
- Most of the texts in the documents are not important for the task

All these observations were verified manually and also using statistical queries and feature and input importance explorations on various working models.

Taking the stated points as assumptions, a new stochastic framework for generating artificial documents was created. The framework allows us to set distribution parameters and generate unique documents by sampling from the generators (for example see 4).

Experiments A series of experiments with increasing difficulty was performed. The experiments featured a simple baseline convolutional model and the model from the first part. The features and metrics were kept as similar to the setting in the first part as possible.

The experiments have confirmed, that even the simple model was able to:

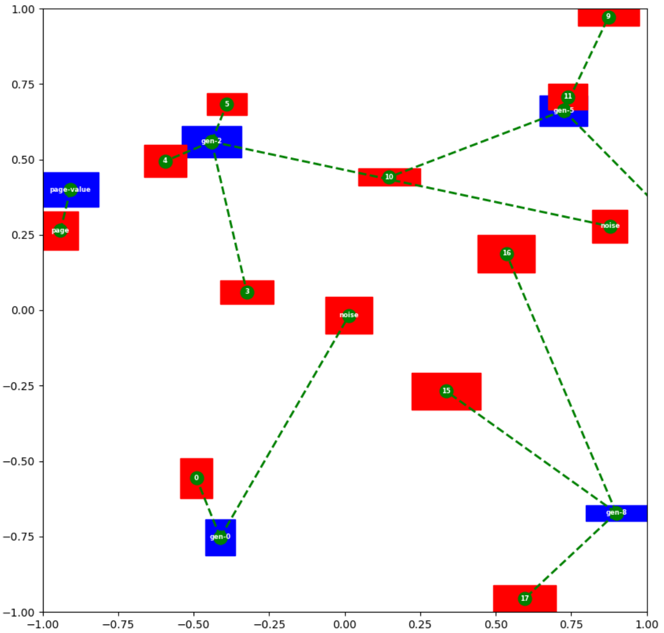
1. learn simple global distributions
2. learn (to 0.9 F_1score) a realistically constructed generator given explicit (“oracle/cheating”) information about which texts are related (‘X explains Y’).
3. learn (to 0.88 F_1 score) the setting of 2) with added shuffling and noise in the form of unimportant word-boxes.

But on the most realistic setting without the oracle/cheating information, the model from the first part failed, as the baseline scored better (0.86) than the tuned model from the first part (0.81).

Conclusion The results have shown, that for generative models, a simple expert approach is not enough to help the training process, as it has failed a check against a simple baseline. On the other hand, it means, that our annotated and well-curated dataset is irreplaceable and valuable. And possibly, that the models (from the first part) can find and exploit information hidden to humans.

This little exploration decides the way forward. The next steps in improving the information extraction system will focus more on exploiting the relations present in the real data.

Figure 4: An example of an artificially generated document and the logic behind it. The blue word-boxes contain the important target information we want to extract and the red ones are generated as the explaining texts, that define the meaning of the blue ones. The resemblance to a real invoice document is, intentionally, just topological.



Using similarity methods for increasing the model's accuracy

The idea behind the usage of similarity related methods is simple - since all the information from one page is already used in the models from the first part, the only way to give the model more information is to add one more similar page.

The framework will function as follows: When predicting the result, it will use already reviewed and annotated pages. This is, in fact, a two-stage process. First, the framework needs to find the useful page from a database by the notions of nearest / most similar search and then use them for extracting information from the new page. The inspiration is drawn from techniques of similarity learning, where, traditionally, siamese networks are used together with triplet loss.

The nearest page search is performed in an embedding space of all the pages, the embeddings are constructed based on visual similarity and taken as a fixed feature.

Baselines The baselines are chosen in such a manner to validate our approach, get more insight into the data and compete with the model from the first part.

- Simple data extraction model (the successful model from the first part with only minor tweaks, see 5).
- Copypaste (templating method - 100% correct for the exact same template).
- Oracle (to quantify the quality of the nearest neighbour search).
- Fully linear (to motivate the use of complex models and to show the nearest neighbour search is not enough with a simple model).

Deep learning similarity-based architectures All the proposed architectures feature a siamese network (5) at the input (from the unknown and the nearest page) and each presents a different approach for combining the information from the unknown (nearest) and known (reference) pages.

- "Triplet Loss architecture" - uses siamese networks in the most 'canonical' way possible with triplet loss.
- "Pairwise classification" - uses a trainable classifier in a pairwise manner over all the combinations of features from reference and nearest page's word-boxes.
- "Query answer architecture" - uses the attention transformer layer as an answering machine to a question of "which word-box has the most similar class to this one".

Nonstandard building blocks that are used are:

- "Tile sequences" - take 2 sequences of items and produce an all-to-all matrix.
- "Pairwise distances" - operate on a matrix of tuples and produce distances.

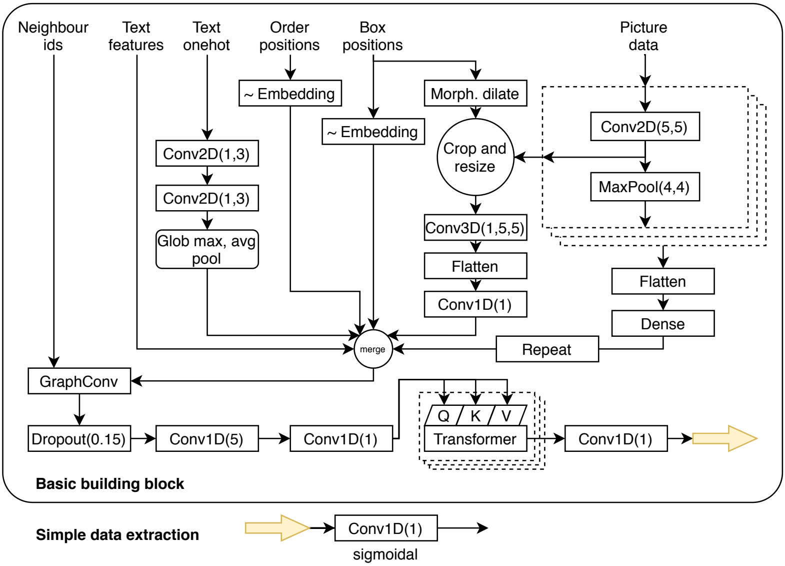


Figure 5: Simple data extraction model. Formally the whole model consists of two parts: a basic building block and a final classification layer. The whole model is formally split into two parts, as the basic building block will be used (as a siamese network) in other models. By removing the final classification layer, we hope to get the best feature representation for each word-box.

- “Filter extracted” - for annotated/nearest page - filter out only word-boxes with nonzero class.
- “Select visible” - for each word-box, get ids of the annotated page - word-boxes that are ‘nearby’. (As if we project the original word-box from unannotated to the nearest page.)
- Distance matrix into ‘triplet loss’ computation layer - sums all contributions from same-class and different class (see ’triplet loss architecture’).

Triplet loss architecture The triplet loss architecture 6 features an extension of the traditional triplet loss (that operates on triplets of datapoints - Reference, Positive example and Negative example):

$$\begin{aligned} L(R,P,N) = \max(\, & \|f(A) - f(P)\|^2 - \\ & \|f(A) - f(N)\|^2 + \alpha, 0) \end{aligned}$$

Two possible triplet loss inspired variants can account for all the datapoints (word-boxes) on a page at once and therefore makes the computation tractable:

$$\begin{aligned} \text{pos\_dist}_{i,j} =& \text{truth\_similar}(i,j) \cdot \text{pred\_dist}(i,j) \\ \text{neg\_dist}_{i,j} =& (1.0 - \text{truth\_similar}(i,j)) \cdot \text{pred\_dist}(i,j) \\ \text{triplet\_like} =& \max_{i,j}(0,\ \alpha + \max(\text{pos\_dist}_{i,j}) \\ & + \min_{i,j}(-\text{neg\_dist}_{i,j})) \\ \text{lossless} =& \sum_{i,j} \text{pos\_dist}_{i,j} - \sum_{i,j} \text{neg\_dist}_{i,j} \end{aligned}$$

Where pos_dist and neg_dist are just helper variables to see the similarity with the original triplet loss, and α is a parameter of the same meaning as in the original triplet loss.

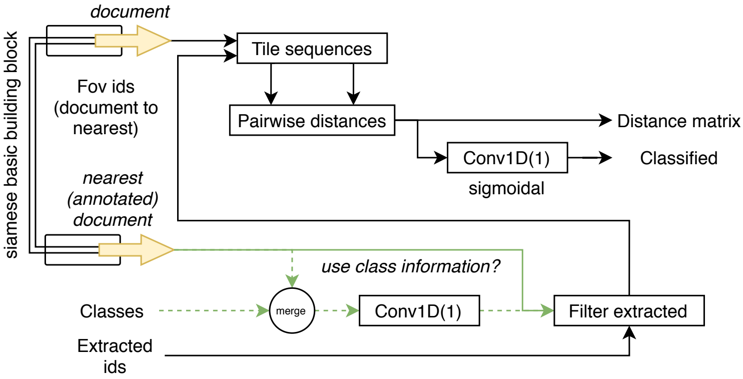


Figure 6: Triplet loss architecture. If we want to add class information from the nearest page, the green dashed version is used.

Pairwise classification architecture The architecture 7 is similar to the previous (triplet loss), but instead of computing the pairwise distances in a feature space, the model concatenates both feature vectors and classifies them into two possible targets (similar class / different class).

At this point it is important to note, that both the triplet loss and pairwise classification architectures are only able to predict classes that are present on the nearest page and to assess their value, their metric is scaled accordingly.

Query answer The query answer architecture 8 is different from the previous two designs in such a way that it has the class information hard-coded in the architecture, which means it can predict a class not present on the nearest page.

Also, this architecture can feature a skip connection to the base information extraction block, take all word-boxes from both the documents at once (for query keys and values) and also feature a field of view information flow to the nearest page.

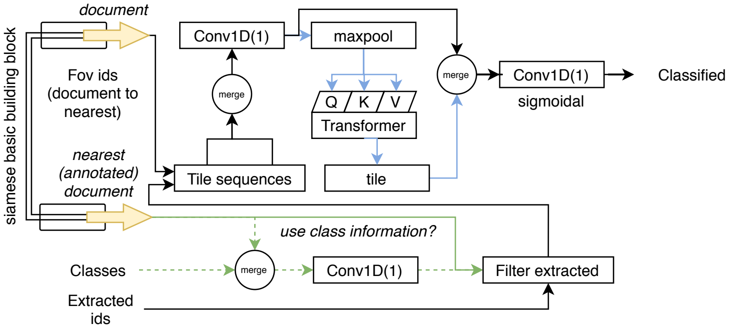


Figure 7: Pairwise classification architecture with an optional refinement module.

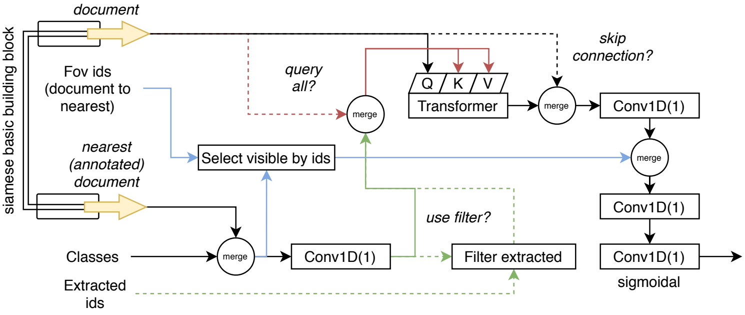


Figure 8: Query answer architecture

Experimental results of the baselines To beat the first part's score we need more than 0.8465 in micro F_1 (which is the score of the model from the previous article tuned for the case without table detection).

The copypaste baseline scored as low as 0.058. Such a low score illustrates the complexity of the task and variability in the dataset - it is not enough to just overlay a different similar known page over the unknown page, as the dataset does not contain completely identical layouts.

The linear baseline scored 0.3085 test micro F_1 score. Even though it did not beat the previous baseline results, it justifies the progress from the basic copypaste model towards more complex trainable architectures with similarity - that a trainable architecture can exploit the similarities.

The results of the oracle baseline illustrate a "moderate quality" of the embeddings – only roughly 60 % of word-boxes have their counterpart (class-wise) in the found nearest page.

Experimental results of the similarity-based architectures Both the pure triplet loss approaches and pairwise classification models performed better than simple copypaste but still worse than linear architecture. The possible reasons could be:

- The existence and great prevalence of unclassified (uninteresting) data in the documents.
- Missing trainable connections to the unknown page.

On the other hand, in the QA architecture, we get a huge improvement of 0.0825 in the F_1 score up to 0.9290. By further analysis, we verify, that the model is versatile enough as the improvement is seen also on the anonymized dataset (by 0.0950) and that all of the visual, geometric and textual features are important for good quality results. Ultimately the ablation study shows, that all the layers and parts of the architecture are important.

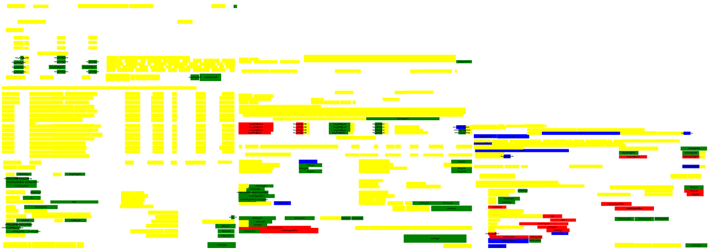


Figure 9: (From left to right) Best classification result of the query answer model – only true positives (green) and true negatives (yellow) can be seen. The worst result of the query answer model. Each blue and red area denotes a mistake. The worst result of the simple data extraction model. Note the minimal count of true positive (green) areas and the dominance of errors (blue and red).

Qualitative comparison of the QA model and the simple data extraction model Ultimately, both models excel at classes that usually appear together - various recipient and sender information. The reason is, that the recipient information is usually required information and as such, it is the most frequent class and therefore it is easy for the network to excel at the detection thereof. Furthermore, the results show, that the previous worst class - page numbering - jumps to a very high score for QA. Moreover, the score for all classes has increased by at least 0.02 points (median gain being 0.04).

A sample of the manual inspection of the results is shown in 9. Both architectures can give a perfect prediction, but the main difference is in the worst samples - the QA model does not make as many errors in the worst cases.

Conclusion

We have explored a novel dataset and successfully replicated some of the deep learning successes in the field of information extraction. We have proved the system distinguishes different tables and that all the parts of the architecture and features we could get from the data are important to get the best results. Ultimately we have designed multiple ways for a deep learning model to incorporate the single-shot learning paradigm into our fully trainable data extraction model.

The QA model has achieved a state-of-the-art performance of 0.9290 micro F_1 score and in practice enables saving of more than thousands of dollars per month.

The dataset was verified by multiple baselines to contain a hard problem unsolvable by other methods. As a part of this research, the dataset is now published in an anonymized version and its size surpasses all the other datasets published in this field to date. The efficient open-source implementation is also published.

Bibliography

Manual typing is expensive: The tco of invoice data capture (part 2). URL https://rossum.ai/blog/manual-typing-is-expensive-the-tco-of-invoice-data-capture-part-2/

Rossum's blogpost "extracting invoices using ai" at medium.com. URL https://medium.com/@bzamecnik/extracting-invoices-using-ai-in-a-few-lines-of-code-96e412df7a7a.

Ahmed Abbasi, Hsinchun Chen, and Arab Salem. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Trans. Inf. Syst.*, 26(3):12:1–12:34, June 2008. ISSN 1046-8188. doi: 10.1145/1361684.1361685. URL http://doi.acm.org/10.1145/1361684.1361685.

Zhili Chen, Liusheng Huang, Wei Yang, Peng Meng, and Haibo Miao. More than word frequencies: Authorship attribution via natural frequency zoned word distribution analysis. *CoRR*, abs/1208.3001, 2012. URL http://arxiv.org/abs/1208.3001.

Bertrand Coüasnon and Aurélie Lemaitre. *Recognition of Tables and Forms*, pages 647–677. Springer London, London, 2014. ISBN 978-0-85729-859-1. doi: 10.1007/978-0-85729-859-1_20. URL https://doi.org/10.1007/978-0-85729-859-1_20.

J. Cowie and W. Lehnert. Information extraction. *Commun. ACM*, 39:80–91, 1996.

V. P. d'Andecy, E. Hartmann, and M. Rusinol. Field extraction by hybrid incremental and a-priori structural templates. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 251–256, April 2018. doi: 10.1109/DAS.2018.29.

Shaveta Dargan, Munish Kumar, Maruthi Rohit Ayyagari, and Gulshan Kumar. A survey of deep learning and its applications: A new paradigm to machine learning. *Archives of Computational Methods in Engineering*, pages 1–22, 2019.

Max Göbel, Tamir Hassan, Ermelinda Oro, and Giorgio Orsi. Icdar 2013 table competition. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1449–1453. IEEE, 2013.

Hatem Hamza, Yolande Belaïd, and Abdel Belaïd. Case-based reasoning for invoice analysis and recognition. In Rosina O. Weber and Michael M. Richter, editors, *Case-Based Reasoning Research and Development*, pages 404–418, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-74141-1.

M. Holecek, A. Hoskovec, P. Baudis, and P. Klinger. Table understanding in structured documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 5, pages 158–164, Sep. 2019. doi: 10.1109/ICDARW.2019.40098.

Martin Holecek. Codes for simple invoice generator, 2019. URL `https://github.com/Darthholi/DocumentConcepts`.

Martin Holecek. Implementation details for this work, source codes and curated anonymized dataset to reproduce results, 2020. URL `https://github.com/Darthholi/similarity-models`.

Martin Holeček. Learning from similarity and information extraction from structured documents. *International Journal on Document Analysis and Recognition (IJDAR)*, pages 1–17, 2021.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

Felix Krieger, Paul Drews, Burkhardt Funk, and Till Wobbe. Information extraction from invoices: A graph neural network approach for datasets with high layout variety. 03 2021.

Zhe Lin and Larry S. Davis. Learning pairwise dissimilarity profiles for appearance recognition in visual surveillance. In *Advances in Visual Computing*, pages 23–34, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-89639-5.

Xiaojing Liu, Feiyu Gao, Qiong Zhang, and Huasha Zhao. Graph convolution for multimodal information extraction from visually rich documents. *arXiv preprint arXiv:1903.11279*, 2019.

Devashish Lohani, Belaïd Abdel, and Yolande Belaïd. An Invoice Reading System using a Graph Convolutional Network. In *International Workshop on Robust Reading*, PERTH, Australia, December 2018. URL `https://hal.inria.fr/hal-01960846`.

David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

HP Newquist. *The Brain Makers, Second Edition.* The Relayer Group, New York, NY, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Pau Riba, Anjan Dutta, Lutz Goldmann, Alicia Fornes, Oriol Ramos, and Josep Llados. Table detection in invoice documents by graph neural networks. pages 122–127, 09 2019. doi: 10.1109/ICDAR.2019.00028.

Maija Tenhunen and Esko Penttinen. Assessing the carbon footprint of paper vs. electronic invoicing. 2010.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv e-prints*, art. arXiv:1706.03762, June 2017.

Oriol Vinyals, Charles Blundell, Timothy P. Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *CoRR*, abs/1606.04080, 2016. URL `http://arxiv.org/abs/1606.04080`.

Peng Wang, Lingqiao Liu, Chunhua Shen, Zi Huang, Anton van den Hengel, and Heng Tao Shen. Multi-attention network for one shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2721–2729, 2017.