Applied mathematics & AI: Neural networks & information extraction

Martin Holeček

Faculty of Mathematics and Physics
Charles University
Department of Numerical Mathematics

Parts of the talk

- Brief intro into neural networks
- The practical problem

Neural networks brief intro

Basic principles and inspirations

Understanding neural networks needs a few ingredients:

- Understanding the origins
  - It is *just* an approximation model
  - Taking inspiration in physiology of neurons in brain
- Basic courses/rules for numerical mathematics
  - Specifically approximation methods - first order gradient descend
- For then doing it yourself - beyond understanding - you then just need
  - Bits of data science
  - And software engineering (at least python. Matlab is for suicidals)

Recap of numerical math [correct me on triggers!]

- You have a function *(to be chosen/described next slide)*
    - that takes various inputs
    - and produces outputs
- You want it to produce different outputs *("***loss / fitness function***")*
    - And you have some parameters of the function to tune *(*called **weights** in NN*)*
- You change the parameters by the law of gradient descend
    - *(This is called* ***backward pass*** *in AI/ML)*
- Given nice properties, the function now produces values closer to what you want
    - Iterate it until sufficiently happy! *(see later)*
- Tons of caveats / needed properties *(this gets you the jobs)*
    - You need a right number of parameters otherwise it cannot get better
    - You need to be sufficiently wise / lucky in the original choice of parameters
    - You can get in depth about the optimizers

Approximation model

The function you optimise is inspired in neurons from the brain,

Each individual **neuron** takes a weighted sum of all its inputs and applies its nonlinear function (usually sigma function).

Since you **stack neurons in layers** and evaluate them all in (one layer in) parallel, this is just vector times matrix of weights, then (vectorised) nonlinearity applied.

Also the model is an universal approximator!

TODO: equation and image

Approximation model

The process of computing the gradients is called the **backward pass** (happens after **forward pass**).

The data science bits

When to stop the learning? (propagating the gradients)

You need the function to produce the desired outputs on a whole dataset you have. Split into [Training set, Validation set, Testing set] and measure how it performs.

At some point it will start to “overfit” on training (= stop predicting meaningfully).

Is that all?

Yes, but

- Different types of layers, architectures, functions
- Different ways to give features, organize training

Notably:

- CNNs
- RNNs
- Transformers
- …

Information Extraction from structured business documents by learning from similarity

Martin Holeček

Faculty of Mathematics and Physics
Charles University
Department of Numerical Mathematics

Outline

- Brief intro into neural networks
- A description of our problem
- Our aims and contributions - the three research questions
- The data
- The metrics
- Feature engineering (the document’s structural information)
- The networks architecture for the first article and its results
- The singleshot learning inspirations
- The baselines
- The inspired architectures
- The results (quantitative and qualitative)

The contents and the links:

Two articles and full source codes and an anonymized dataset (of 25000 documents):

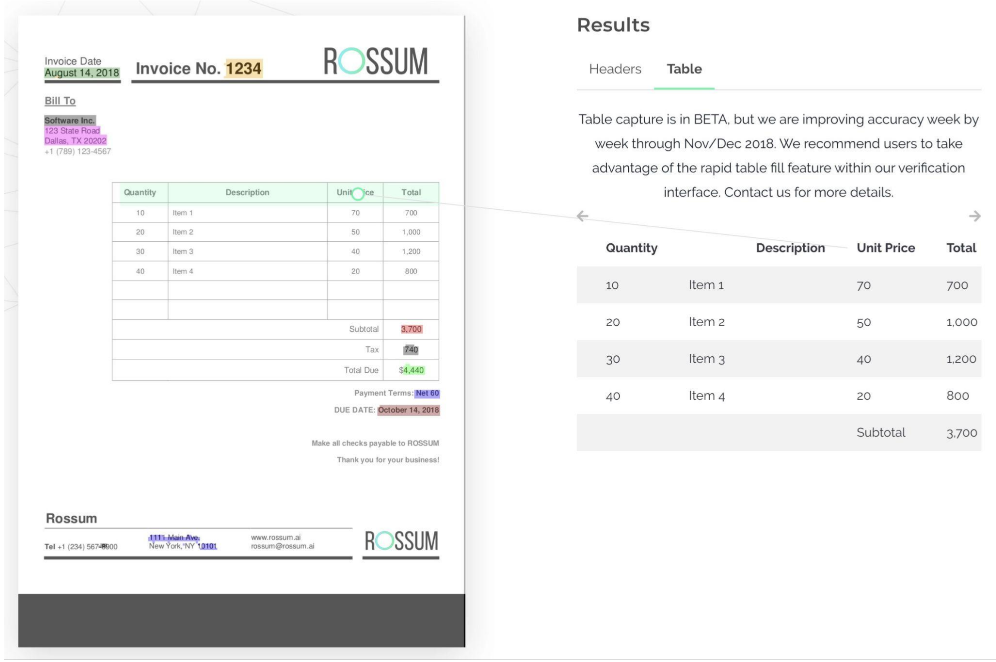
- https://arxiv.org/abs/1904.12577
- https://arxiv.org/abs/1904.12577
- https://github.com/Darthholi/similarity-models
- https://github.com/Darthholi/DocumentConcepts

The main enablers of this research:

- The information extraction task at the hearth of document automation
- A novel, huge, curated dataset
- Methods (and hardware) moved times of “AI Winter” -> deep learning

The task

INVOICE
ISSUE DATE
27 July 2018
INVOICE NO
123456789
ROSSUM
Ship to:
ROSSUM
1234 North Street
London SW7 2AP UK
+44 20 5670 8901
rossum@rossum.ai
Bill to:
ABC Company
7890 South Street
Manchester M1 1BE UK
+44 616 234 5678
email@email.co.uk
ORDER NUMBER
910101
PAYMENT TERMS
Due on Receipt
DATE DUE
1 August 2018
QUANTITY
DESCRIPTION
UNIT PRICE
AMOUNT
150 Item 1 £15 £2250
100 Item 2 £30 £3000
35 Item 3 £45 £1575
25 Item 4 £100 £2500
Subtotal 9.325
Sales Tax 1.865
Total (GBP) 11.190£



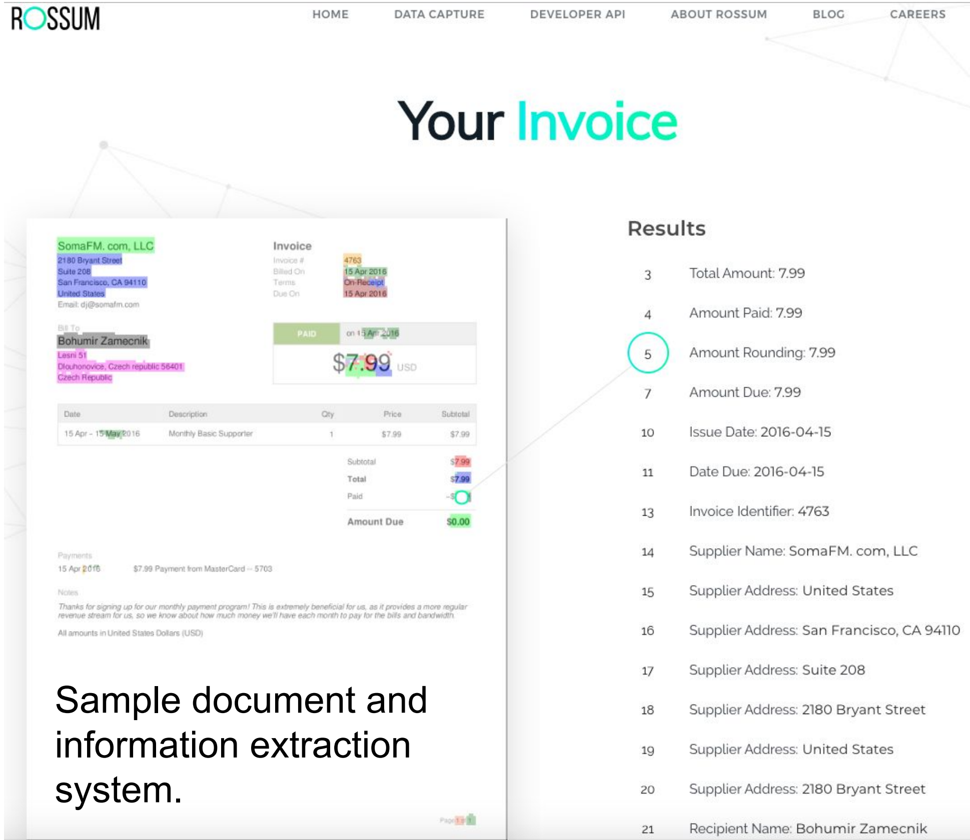
Invoice Date
August 14, 2018
Invoice No. 1234
ROSSUM
Bill To
Software Inc.
123 State Road
Dallas, TX 20202
+1 (789) 123-4567
Quantity
Description
Total
10 Item 1 70 700
20 Item 2 50 1,000
30 Item 3 40 1,200
40 Item 4 20 800
Subtotal 3,700
Tax 740
Total Due $4,440
Payment Terms: Net 60
DUE DATE: October 14, 2018
Make all checks payable to ROSSUM
Thank you for your business!
Rossum
www.rossum.ai
rossum@rossum.ai
Results
Headers
Table
Table capture is in BETA, but we are improving accuracy week by week through Nov/Dec 2018. We recommend users to take advantage of the rapid table fill feature within our verification interface. Contact us for more details.
Quantity
Description
Unit Price
Total
10 Item 1 70 700
20 Item 2 50 1,000
30 Item 3 40 1,200
40 Item 4 20 800
Subtotal 3,700

The information extraction task and motivation

- Texts - individual words in business documents.
- The targeted information = classification of the texts that helps in automation.

Medium-sized company:

- ~25k invoices per month
- 1 % improvement ~ 500$ savings/monthly

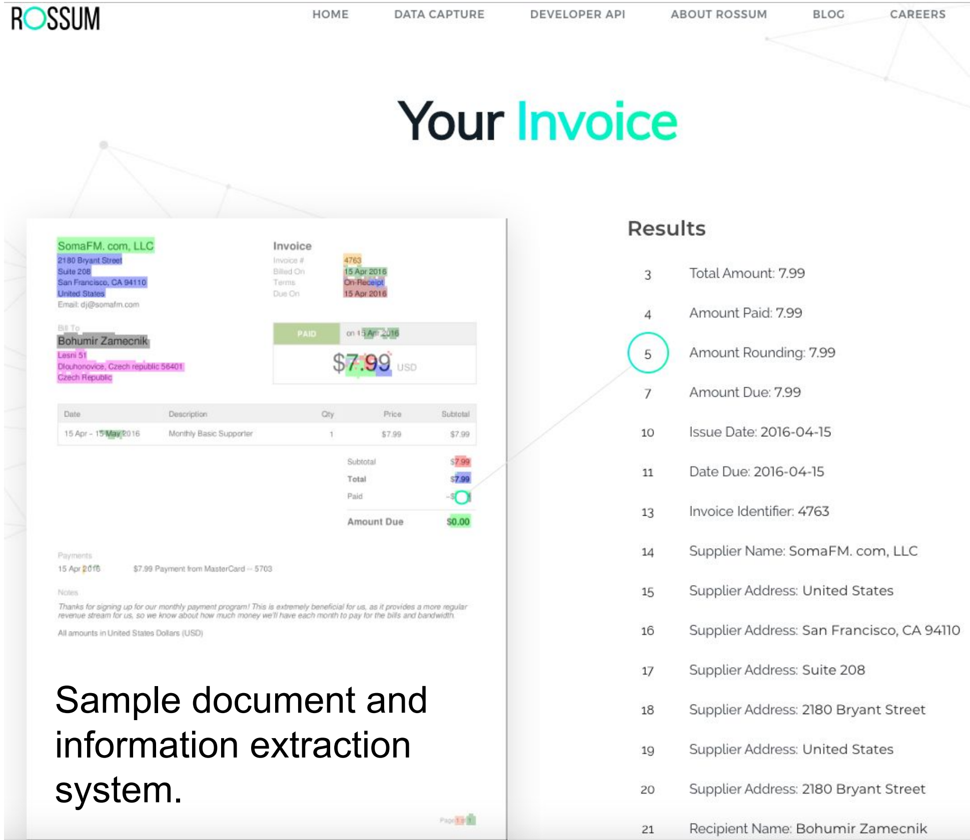


Sample document and information extraction system.

The structured documents

We work with structured documents, where not only the textual content, but also the positioning matter (no fixed set of layouts actually exists!)

We want to extract important information like address, date, id details, amount types, tax details, … (35 classes total)



Sample document and information extraction system.

The aim & our contribution

1) We work on unexplored dataset of **invoices** and **business documents**
   - We have published an anonymized version!
   - Bigger than any other work was using -> therefore supposedly allows for deep learning
2) Answering the question whether one “end to end” fully trained (free of heuristic reasoning) model is able to succeed at information extraction
   - Exploration of the importance of inputs (‘ablation’), different architecture modules, comparing against a baseline, uing convolution, Graph CNN, self-attention layers used at once…
   - No referenced paper or commercial solution can be customized to fit our aim, so we present our method as a novel approach and compare only against logistic regression baseline
3) Exploring the idea of a “single shot learning paradigm” and its added value
   - designing multiple architectures and testing their parts
   - Publishing the code and anonymized dataset.

First part:
A model that distinguishes between various information in a page full of structures, tables and images

The journey - what was tried before

- Multiple combined methods and heuristics for table detection, starting with detecting tables based on layouts or graphical borders
- All those have failed, because we desired a fully trainable system not limited by specific document features
- For example for trainable table detection/extraction we have tried purely graphical methods like R-CNN variants and YOLO, they did not work well in our case

The answer is then to try to exploit all the information we can in a data driven model.

*(Ad 1: We work on unexplored dataset of **invoices** and **business documents**)*

The data - annotated structured documents

- Annotation process of trained professionals with another human supervisor and automatic corrections (like overlapping thresholds)
- Total size: 25071 PDFs
  - All the documents are business documents like invoices
  - We have excluded OCR’d documents to not measure a joint performance of OCR+our method
- The splits are ¾ training, ¼ validation and test

Each document has around 500 word-boxes per page and 2 pages on average.

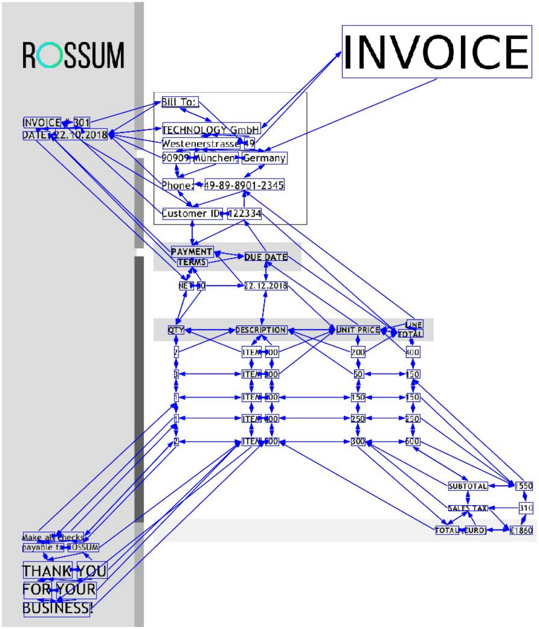
(Ad 1: We work on unexplored dataset of ***invoices*** *and* ***business documents)***

The metrics of measuring the success

- Operating on **word-boxes**
  - each can be labelled with a line-item table class or other 37 classes.
- Metrics aggregation method chosen is ‘micro’
- Line-items will be evaluated with F_1 score
  - (harmonic mean of precision and recall)
- Other classes with F_1 scores on positive samples
  - Unbalanced labels problem (only 1.2 % positive labels)
- Not evaluating against methods from other papers
  - Using logistic regression baseline
- Metric similar to the one used at ICDAR competition (we use wordboxes, they used charboxes - individual letters)

Feature engineering: the structured information

- Geometrical
  - Reading order of word-boxes (ordering of inputs; order for positional embedding)
  - Neighbouring ‘seen’ word-boxes (for graph convolution)
  - Coordinates (for positional embedding)
- Textual
  - Trained embedding
  - Features for capturing named entities
- Image features
  - Whole image (for convolutional layers)
  - Crops around each word-box



Features for each wordbox are then fed into the network.

Note: All these features can be perturbed during training for regularizations.

How do we usually work with words in AI? [NLP]

You need to turn text into numbers somehow (keywords: “tokenisation, …”)

- Ranging from each word (or character) having its own index
- Or mapping a word into a space of meanings (embeddings, pretrained in a language)

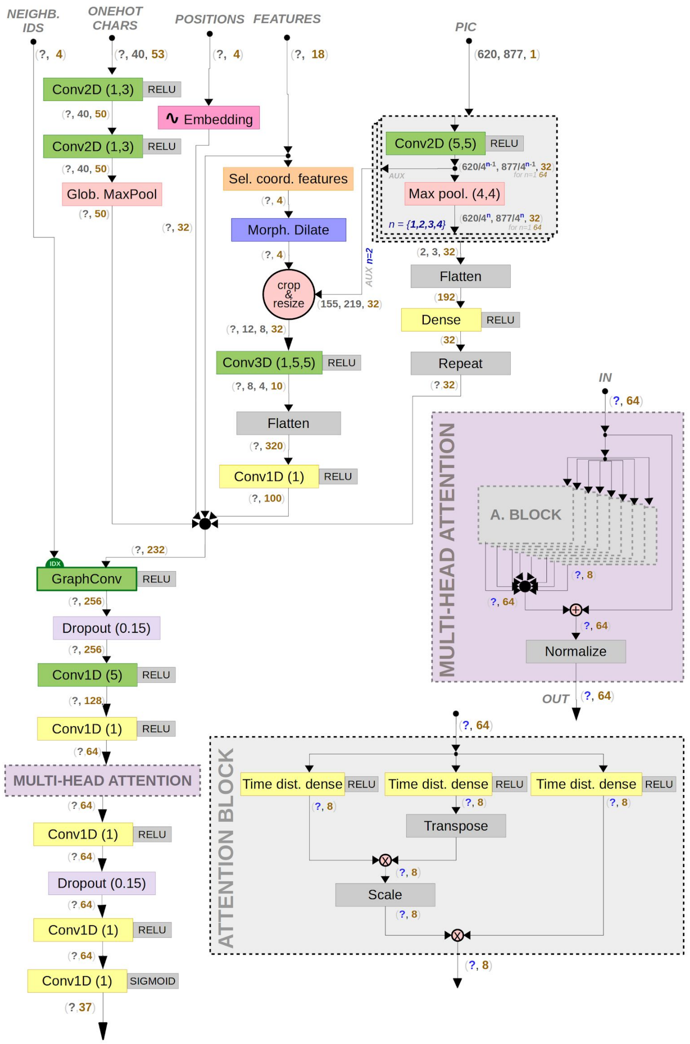
The problems include working with named entities and identification and finding the best method that works for your case.

We did it by using multiple representations at the same time!

(Ad 2: Can one model succeed at information extraction?)

The architecture

- Concatenation of all features before blocks of:
  - Blocks of Graph CNN (over neighbours)
  - CNN over sequence ordering
  - Multi-head attention
- Architecture is a result of optimizations, so we will provide experiments on:
  - Exploration of the importance of inputs, different architecture modules and comparing against a baseline
- Sigmoidal & binary crossentropy as loss
- 867281 trainable (float) parameters
- ‘Adam’ optimizer



Results - number of neighbours & the baseline

- Logistic regression baseline improves with neighbours but fails to generalize ‘others’ classes
- Finding line-items table is ‘easier’ than finding other structural information.
    - Possibly due to class imbalance (even for a human)
- More neighbours can improve the line-items table body finding score

Experiments against the baseline	Adaptation			Generalization		
	line-items		others	line-items		others
	body F_1	header F_1	micro F_1	body F_1	header F_1	micro F_1
complete model (without neighbours)	0.9666	0.9969	0.8687	0.9242	0.9876	0.6609
complete model (1 neighbour)	0.9738	**0.9967**	**0.8790**	0.9389	**0.9864**	**0.6650**
complete model (with 2 neighbours)	**0.9762**	0.9963	0.8749	**0.9408**	0.9860	0.6629
logistic regression without neighbours	0.7594	0.9477	0.0004	0.7560	0.9362	0.0000
logistic regression with 1 neighbour	0.8664	0.9663	0.1482	0.8071	0.9461	0.0327
logistic regression with 2 neighbours	0.8939	0.9724	0.2276	0.8284	0.9493	0.0525

Results - ablation

- Table header finding might perform better with focal loss (does comply with theory, as they are a smaller class than table body)
- Attention is required for better generalization
- Sequence convolution is important
    - Allows reading the text in order, but also to notice beginnings/endings

Experiments with ablation	Adaptation			Generalization		
	line-items		others	line-items		others
	body F_1	header F_1	micro F_1	body F_1	header F_1	micro F_1
complete model	0.9738	0.9967	0.8790	**0.9389**	0.9864	**0.6650**
focal loss	0.9735	**0.9969**	0.8557	0.9383	**0.9878**	0.6398
no convolution over sequence	0.9670	0.9945	0.8638	0.9101	0.9800	0.6237
no attention	**0.9780**	0.9967	**0.8806**	0.9348	0.9864	0.6487
no convolution with dropout after attention	0.9646	0.9950	0.8435	0.9168	0.9807	0.6050

Results - input and dataset variations

- Anonymized score is not zero (so mutual positional information is important!)
- Even the basic text features help the model generalize well.
- Although the model has been optimized on the smaller dataset, it has the capacity to work nicely on bigger datasets.

Experiments with inputs variations	dataset	Adaptation			Generalization		
		line-items		others	line-items		others
		body F_1	header F_1	micro F_1	body F_1	header F_1	micro F_1
complete model (all inputs)	**small**	**0.9738**	**0.9967**	**0.8790**	**0.9389**	**0.9864**	**0.6650**
no text embeddings	small	0.9702	0.9921	0.7772	0.9108	0.9771	0.5118
no picture, only some text features	anonym	0.9694	0.9943	0.4518	0.9185	0.9805	0.4745
no picture, no text features	anonym	0.9588	0.9848	0.6836	0.8919	0.9549	0.2152
complete model (all inputs)	big	N/A	N/A	0.8487	N/A	N/A	N/A

Results - target variations

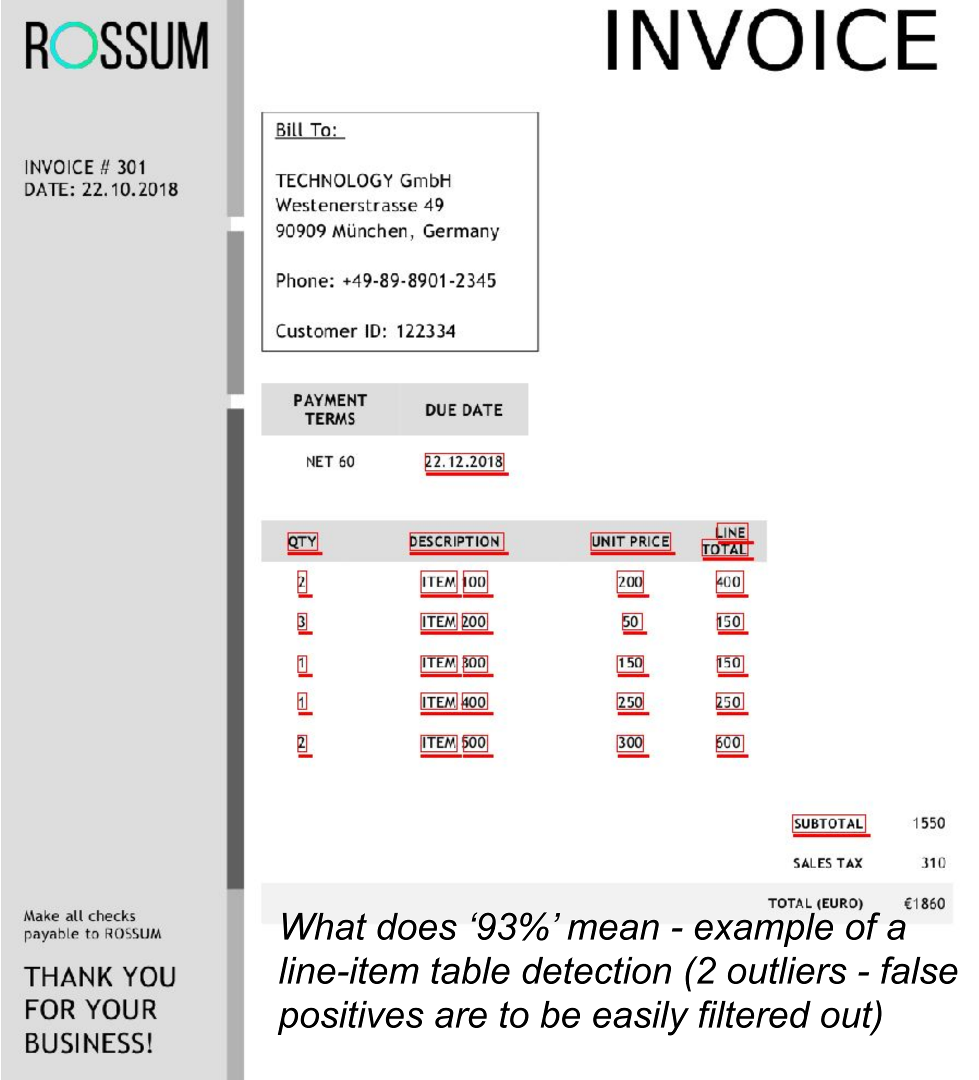
- The tasks of finding line-items and other structural information do boost each other
- Omitting the need to classify line-item table header leads to higher generalization score

Experiments with training target variations	dataset	Adaptation			Generalization		
		line-items		others	line-items		others
		body F_1	header F_1	micro F_1	body F_1	header F_1	micro F_1
complete model (all outputs)	**small**	**0.9738**	**0.9967**	**0.8790**	0.9389	**0.9864**	0.6650
only line-items	small	0.9027	0.9950	N/A	0.8762	0.9766	N/A
no line-item header	small	0.9736	N/A	0.8777	**0.9394**	N/A	**0.6731**
all but line-items	small	N/A	N/A	0.8632	N/A	N/A	0.6247
complete model (other than line-items targets)	big	N/A	N/A	0.8487	N/A	N/A	N/A

(Ad 2: Can one model succeed at information extraction?)

2nd question conclusion

- Detecting **line-items class**: 93 % (special testset)
- Information extraction:
  - **other classes**: 87 % (layouts with similar parts)
  - **other classes**: 66 % (completely different layouts)
- Adapts to bigger datasets; Information extraction and line-item table detection targets do boost each other;
- Synergy of GNN, convolutions & global self-attention


ROSSUM

INVOICE

INVOICE # 301
DATE: 22.10.2018

Bill To:

TECHNOLOGY GmbH
Westenerstrasse 49
90909 München, Germany

Phone: +49-89-8901-2345

Customer ID: 122334

PAYMENT TERMS	DUE DATE
NET 60	22.12.2018

QTY	DESCRIPTION	UNIT PRICE	LINE TOTAL
2	ITEM 100	200	400
3	ITEM 200	50	150
1	ITEM 300	150	150
1	ITEM 400	250	250
2	ITEM 500	300	600

SUBTOTAL	1550
SALES TAX	310
TOTAL (EURO)	€1860

Make all checks payable to ROSSUM

THANK YOU FOR YOUR BUSINESS!

What does '93%' mean - example of a line-item table detection (2 outliers - false positives are to be easily filtered out)

Final part:
Deep learning, siamese networks and similarity to improve the extraction score even further

(table detection from the first part discontinued)

Single-shot learning and inspirations

Already all the information from one page is used in the model.

Let's add one more "similar" page!

- When predicting the result, we are able (or even required) to use (already reviewed/annotated) items.
- A two stage process:
  - Find the useful item(s) from a database ('nearest/most similar search')
  - Use them for predicting the new item
- Tight relationship with techniques of similarity learning

Traditionally, Siamese networks are used with triplet loss in this task.

(Recap) Common ground with the first part

- Same dataset with 25071 business documents and training split.
  - Each document has around 500 wordboxes per page and 2 pages on average.
- Metric operating on wordboxes
  - Each will be labelled with a class (or marked to not be extracted).
- Metrics aggregation method chosen is ‘micro F_1 score’
  - Unbalanced labels problem (only 1.2 % positive labels)

… And a note on differences:

- Here we focus only on extracting specific information and not whole tables.
  - While proving, that the system from the previous paper is versatile enough.
- We will measure generalization while having access to similar documents.
- Previous validation set is split into new validation and new test set.
- We are working on “the bigger” dataset only.

Our singleshot designs - common concepts

- Fixed nearest page search (based on visual embeddings)
  - taken as a fixed feature, not a scope of this work to improve them
- Each document's page can select the nearest annotated page only from the previous documents in a given ordering.
  - As in a real service we can only see the already arrived and processed.
- We want the method to be robust and so before each epoch, the order of all pages would be shuffled and only the previous (in the said order) pages from a different document are allowed to be selected.
  - This holds for all sets (training, validation and test) separately. In a practical application, we could make the inference perform even better by allowing it to use (for example) the train set as a datasource for the 'nearest annotated' input.

Baselines

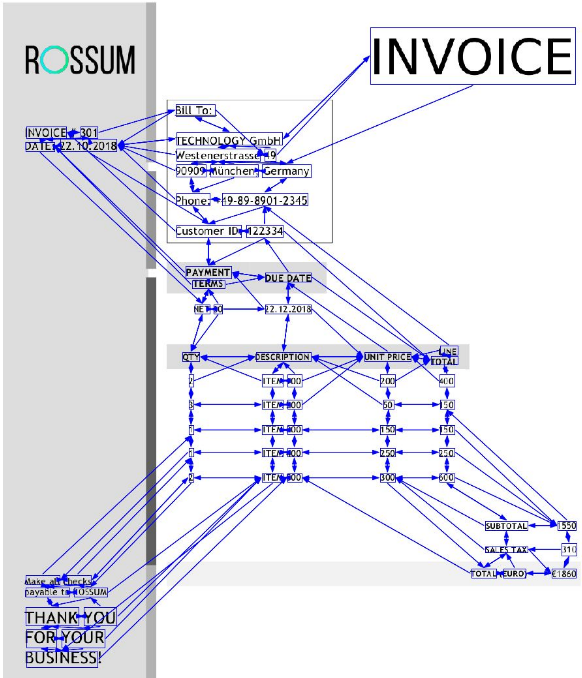
- Simple data extraction model
  - = The successful model from question 2 (the first article), see next slide
- Copypaste
  - Templating method - 100% correct for the exactly same template
- Oracle
  - To quantify the quality of the nearest neighbour search.
- Fully linear
  - To motivate the use of complex models and to show the nearest neighbour search is not enough with a simple model

(Recap) The whole structured information is used

- Geometrical
  - Reading order of word-boxes (ordering of inputs; order for positional embedding)
  - Neighbouring ‘seen’ word-boxes (for graph convolution)
  - Coordinates (for positional embedding)
- Textual
  - Trained embedding
  - Features for capturing named entities
- Image features
  - Whole image (for convolutional layers)
  - Crops around each word-box

Features for each wordbox are then fed into the network.

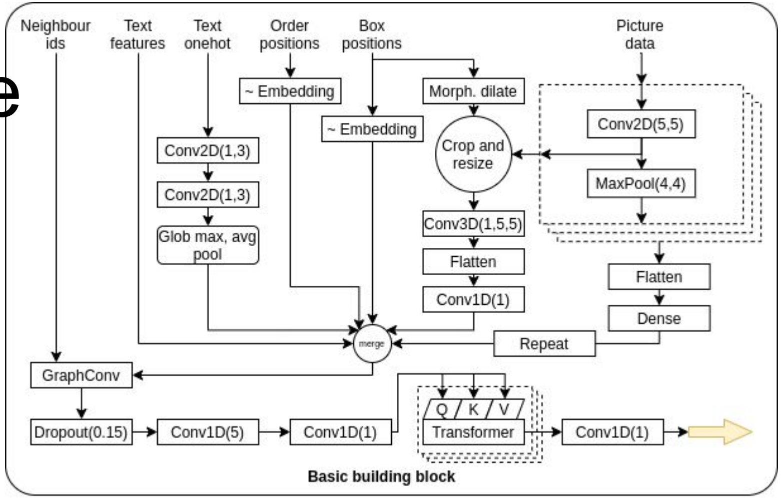
Note: All these features can be perturbed during training for regularizations.



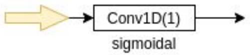
(Recap) The shared architecture

- Concatenation of all features, then blocks of:
    - Graph CNN (over neighbours)
        - Local dependencies
    - CNN over sequence ordering
        - Reading order
    - Multi-head attention (Transformer)
        - Global layouts
- Sigmoidal & binary crossentrophy as a loss
- ~900000 trainable (float) parameters
- ‘Adam’ optimizer

+ Minor (hyperparam) differences to our previous work, overall the same validated model.



Simple data extraction



Note, that we will call a part of the baseline model ‘Basic building block’ and we will use it in the singleshot learning designs as a siamese part at the input.

Architecture choices

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

Building blocks and used ideas in the models

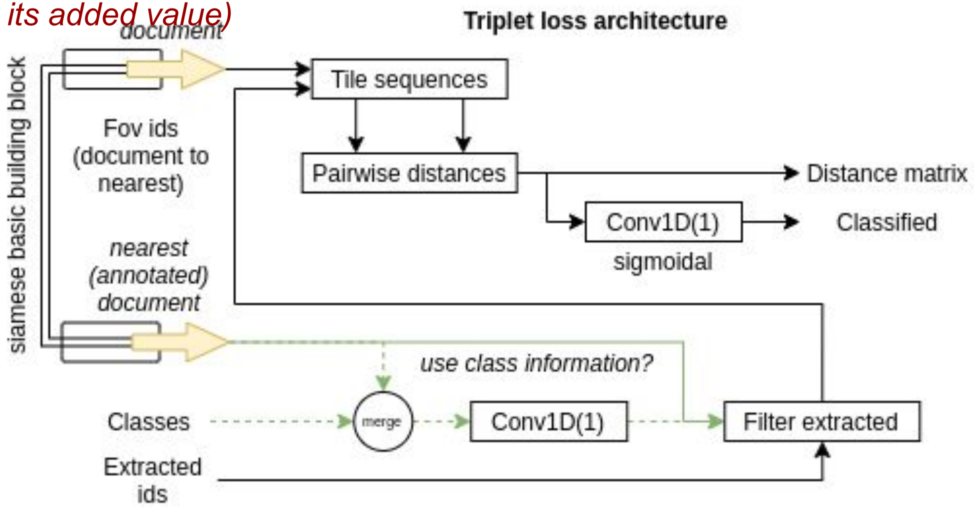
- “Tile sequences” - take 2 sequences of items and produce all2all matrix
- “Pairwise distances” - operate on matrix of tuples and produce distances
- “Filter extracted” - for annotated/nearest page - filter out only wordboxes with nonzero class
- “Select visible” - for each word-box, get ids of annotated page - word-boxes that are ‘nearby’
    - (as if we project the original wordbox from unannotated to the nearest page)
- Distance matrix into ‘triplet loss’ computation - sums all contributions from same-class and different class (see next slide)

(Ad 3: Exploring the idea of a “single shot learning paradigm” and its added value)

Triplet Loss architecture

Options:

- Add annotated class information to the nearest page's features.
- Use a triplet-like loss with [different type]() of a constraint.
- Modifying the distance computation so that the model has the ability to use not only euclidean space, but also a cosine similarity.



$$L(R, P, N) = \min(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha, 0)$$

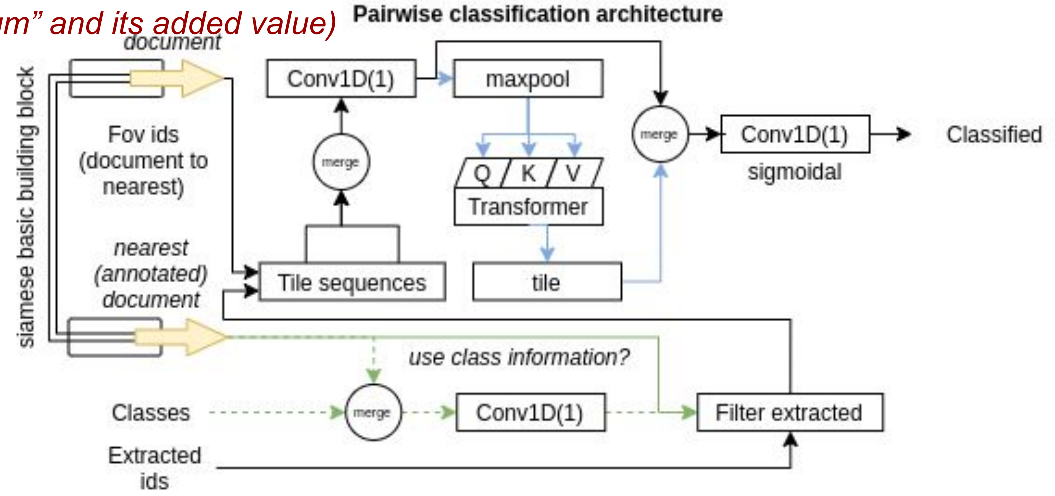
Traditional triplet loss (Reference, Positive, Negative)

triplet_like/lossless variants for more word-boxes

$$\begin{aligned}
\text{pos_dist}_{i,j} &= \text{truth_similar}(i,j) \cdot \text{pred_dist}(i,j) \\
\text{neg_dist}_{i,j} &= (1.0 - \text{truth_similar}(i,j)) \cdot \text{pred_dist}(i,j) \\
\text{triplet_like} &= \max(0,\ \alpha + \max(\text{pos_dist}_{i,j}) \\
&\quad + \min(-\text{neg_dist}_{i,j})) \\
\text{lossless} &= \textstyle\sum_{i,j} \text{pos_dist}_{i,j} - \sum_{i,j} \text{neg_dist}_{i,j}
\end{aligned}$$

(Ad 3: Exploring the idea of a "single shot learning paradigm" and its added value)

Pairwise classification



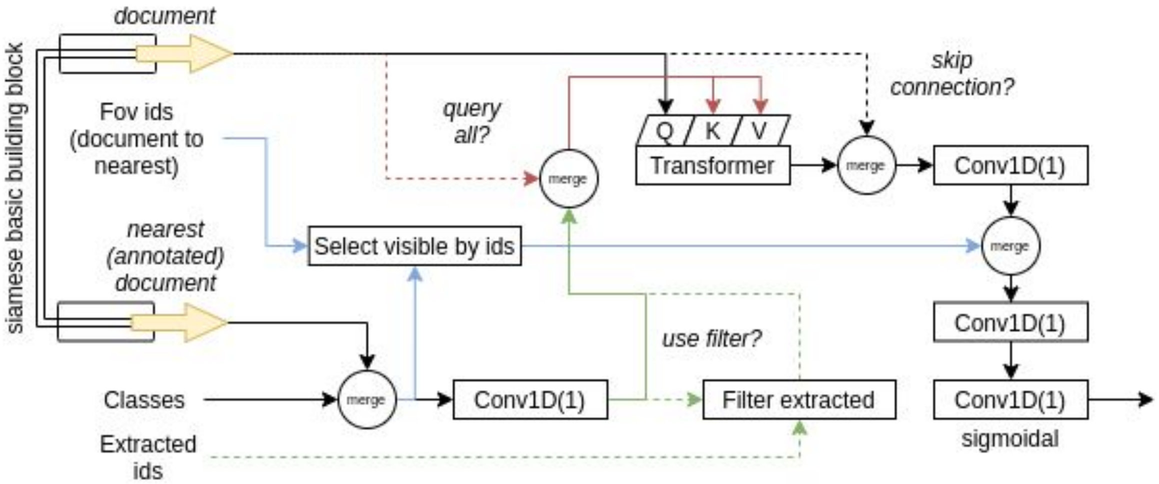
Options:

- (like before)
- Feature a global refinement section, that pools information from each wordbox, uses a global transformer and propagates the information back to each reference wordbox - nearest wordbox pair to be classified once more with the refinement information.

(Ad 3: Exploring the idea of a "single shot learning paradigm" and its added value)

Query Answer

Query answer architecture



Options:

- Query keys and values to be only from the nearest page, or all wordboxes from both documents?
- Feature a skip connection to the base information extraction block?
- Filter only annotated nearest page's wordboxes? (As in the two previous approaches)
- Add a field of view information flow to the nearest page for each wordbox?

Baseline results: Simple data extraction model (prev)

To beat the previous article’s score we need > 0.8465 F1

Model from previous article tuned for the case (without some previous article specific details such as table detection etc…)

Simple data extraction - test micro F1 score

2x attention layer, feature space 640	0.6220
1x attention layer, feature space 640	0.8081
1x attention layer, feature space 64	**0.8465**
1x attention layer, f. space 64, fully anonymized	0.6128
1x attention layer, f. space 64, only text features	0.7505

Baseline results: Copypaste (templating)

Such a low score illustrates the complexity of the task and variability in the dataset. Simply put, 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.We can also see that an important consistency principle for the nearest neighbors holds:

- Selecting a random page decreases the score.
- Using a bigger search space for the nearest page increases the score.

Nearest page by embeddings and from validation set (standard)	0.0582
Nearest page search from validation and train set	0.0599
Nearest page set to random	0.0552

Baseline results: Oracle and linear

Linear baseline

- Scored **0.3085** test micro F1 score.
  - Justifies the progress from the basic Copypaste model towards trainable architectures with similarity.
- Does not beat the previous baseline results
  - Proves that the similarity principle alone is not sufficient; justifies the design of more complicated models.

Oracle

“moderate quality” of the embeddings – only roughly 60% of word-boxes have their counterpart (class-wise) in the found nearest page.

Oracle setting	Hits
Nearest page by embeddings and from validation set (standard)	59.52 %
Nearest page search from validation and train set	60.43 %
Nearest page set to random	60.84 %

Triplet loss & Pairwise classification - results

Both pure triplet loss approaches and pairwise classification performed better than simple Copypaste, but still worse than linear architecture.

Reasons:

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

Triplet loss - test micro F1 score

1x attention layer, loss-less variant	0.0619
2x attention layer, loss-less variant	0.0909
1x attention layer	0.1409
2x attention layer	0.1464

Pairwise - test micro F1 score

2x attention layer + refine section	0.2080
2x attention layer	0.2658
1x attention layer	0.2605

QA - results (the best model)

- We get a huge improvement of 0.0825 in the F1 score wrt to the baseline
- Versatile enough - improvement is seen also on anonymized dataset (by 0.0950).
  - It also verifies that all of the visual, geometric and textual features are important for good quality results.

Ablation study shows, that all the layers and parts are important.

Architecture - test micro F1 score

All QA improvements in place	**0.9290**
Fully anonymized dataset	0.7078
Only text features	0.8726
Nearest page set to random	0.8555
Without field of view	0.8957
Without query all	0.7997
Without skip connection	0.9002
Without filtering	0.8788

Qualitative comparison

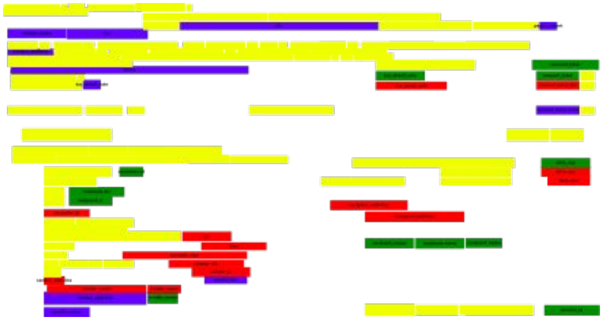
- Both models excel at classes that usually appear together - various recipient and sender information.
- Recipient information is usually a required information -> the most frequent class ->and so it is easy for the network to excel at the detection thereof.

Previous worst class - page numbering - jumps to a very high score for QA.

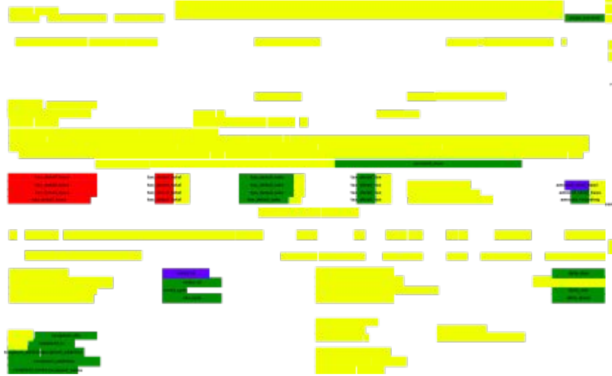
Moreover the score for all classes has increased by at least 0.02 points (median gain being 0.04).

Best and worst performing fields (and their scores)	Simple – test micro F_1 score	QA – test micro F_1 score
Worst classes of Simple data extraction model		
Page current	0.30	0.90
Page total	0.35	0.88
Terms	0.62	0.78
Best classes of Simple data extraction model		
Recipient DIC	0.94	0.96
Recipient IC	0.94	0.97
Spec Symbol	0.94	0.96
Worst classes of Query answer		
Order ID	0.65	0.75
Terms	0.62	0.78
Customer ID	0.75	0.83
Best classes of Query answer		
Sender IC	0.93	0.96
Spec Symbol	0.94	0.96
Recipient IC	0.94	0.97

Comparison of results (QA vs baseline: prev. model)



Previous worst



QA - worst

QA - best

Legend:

Yellow = true negative
Green = true positive

Red = false positive
Blue = false negative

Conclusion

- Successfully replicated the deep learning successes on a novel dataset
- Proved the system distinguishes different tables and makes use of all layers
- Designed multiple ways for a deep learning model to incorporate single shot learning paradigm into our fully trainable data extraction model.
- The dataset was verified by multiple baselines to contain a hard problem
- QA model: successful improvement of 8.25% in F1 score (~ thousands $/month)
- All parts of the architecture are important to get the results
- Improvement of previously underperforming classes, all classes strictly better
- Publication of the greatest anonymized dataset of documents (to date)
- Published efficient open-source implementation (trained in <4 days on one GPU)

Questions

- Your questions
- Specific topics in details (how convolutipns work, how rnn work)

Thank You!

martin.holecek.ai@gmail.com

codes: https://github.com/Darthholi/similarity-models