Utilization of contextual information for post-OCR error correction using language models

PV212

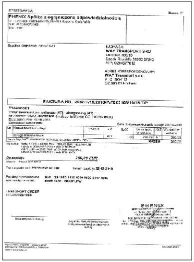
Dávid Meluš
485455@mail.muni.cz

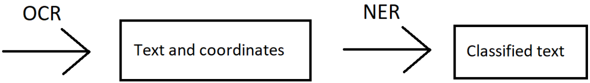
Fakulta informatiky Masarykovej univerzity

23. 3. 2023

Motivation

IBO project, `https://www.fi.muni.cz/app/projects?project=64989`





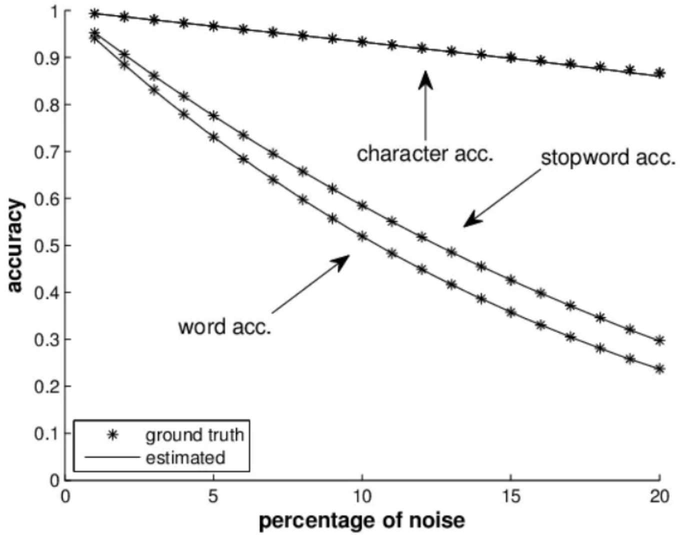
Current OCR

EasyOCR, `https://github.com/JaidedAI/EasyOCR`

- further fine-tuned on born-digital dataset of invoices
- fast, one of the faster ocr models
- easy to use

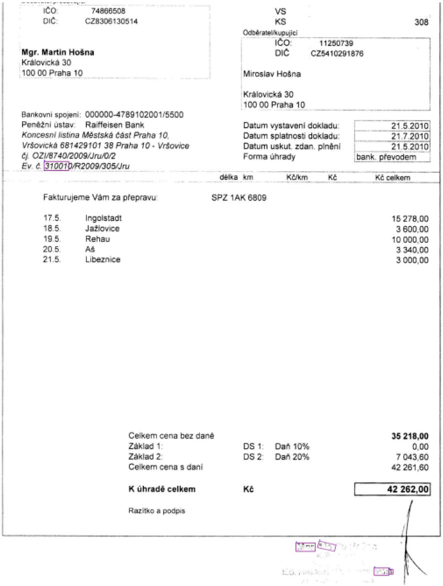

```
[([[189, 75], [469, 75], [469, 165], [189, 165]], '愚园路', 0.3754989504814148),
 ([[86, 80], [134, 80], [134, 128], [86, 128]], '西', 0.40452659130096436),
 ([[517, 81], [565, 81], [565, 123], [517, 123]], '东', 0.9989598989486694),
 ([[78, 126], [136, 126], [136, 156], [78, 156]], '315', 0.8125889301300049),
 ([[514, 126], [574, 126], [574, 156], [514, 156]], '309', 0.4971577227115631),
 ([[226, 170], [414, 170], [414, 220], [226, 220]], 'Yuyuan Rd.', 0.8261902332305908),
 ([[79, 173], [125, 173], [125, 213], [79, 213]], 'W', 0.9848111271858215),
 ([[529, 173], [569, 173], [569, 213], [529, 213]], 'E', 0.8405593633651733)]
```


```
[([[71, 49], [489, 49], [489, 159], [71, 159]], 'ポ<捨て禁止!', 0.6339447498321533),
 ([[95, 149], [461, 149], [461, 235], [95, 235]],
  'NOLITTER',
  0.32493865489959717),
 ([[80, 232], [475, 232], [475, 288], [80, 288]],
  '清潔できれいな港区を',
  0.9784268140792847),
 ([[109, 289], [437, 289], [437, 333], [109, 333]],
  '港 区 MINATO CITY',
  0.18788912892341614)]
```



accuracy
percentage of noise
character acc.
stopword acc.
word acc.
ground truth
estimated
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
20

Errors example



IČO: 74866508
DIČ: CZ8306130514

Mgr. Martin Hošna
Královická 30
100 00 Praha 10

VS
KS 308

Odběratel/kupující
IČO: 11250739
DIČ: CZ5410291876

Miroslav Hošna

Královická 30
100 00 Praha 10

Bankovní spojení: 000000-4789102001/5500
Peněžní ústav: Raiffeisen Bank
Koncesní listina Městská část Praha 10,
Vršovická 681429101 38 Praha 10 - Vršovice
čj. OZI/8740/2009/Jru/0/2
Ev. č. 310010/R2009/305/Jru

Datum vystavení dokladu:	21.5.2010
Datum splatnosti dokladu:	21.7.2010
Datum uskut. zdan. plnění	21.5.2010
Forma úhrady	bank. převodem

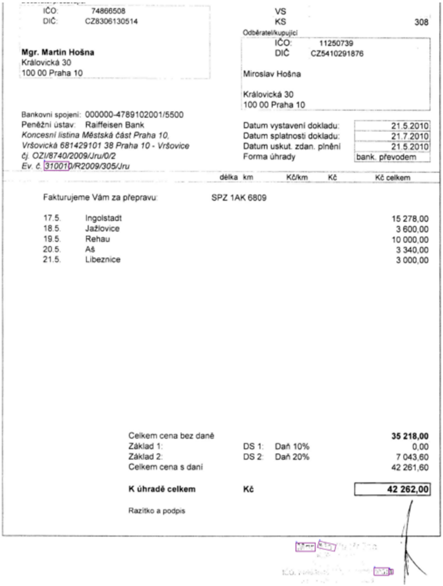
		délka km	Kč/km	Kč	Kč celkem
Fakturujeme Vám za přepravu:	SPZ 1AK 6809				
17.5.	Ingolstadt				15 278,00
18.5.	Jažlovice				3 600,00
19.5.	Rehau				10 000,00
20.5.	Aš				3 340,00
21.5.	Libeznice				3 000,00

Celkem cena bez daně			**35 218,00**
Základ 1:	DS 1:	Daň 10%	0,00
Základ 2:	DS 2:	Daň 20%	7 043,60
Celkem cena s daní			42 261,60
K úhradě celkem	**Kč**		**42 262,00**

Razítko a podpis

```
confidence: 0.28953    prediction: 31001s,
confidence: 0.10326    prediction: aa*,
confidence: 0.04340    prediction: 2ia:',
confidence: 0.06670    prediction: &,
confidence: 0.16727    prediction: B*-,1,
```

Errors example



IČO: 74866508
DIČ: CZ8306130514

Mgr. Martin Hošna
Královická 30
100 00 Praha 10

VS
KS 308

Odběratel/kupující
IČO: 11250739
DIČ: CZ5410291876

Miroslav Hošna

Královická 30
100 00 Praha 10

Bankovní spojení: 000000-4789102001/5500
Peněžní ústav: Raiffeisen Bank
Koncesní listina Městská část Praha 10,
Vršovická 681429101 38 Praha 10 - Vršovice
čj. OZI/8740/2009/Jru/0/2
Ev. č. 310010/R2009/305/Jru

Datum vystavení dokladu:	21.5.2010
Datum splatnosti dokladu:	21.7.2010
Datum uskut. zdan. plnění	21.5.2010
Forma úhrady	bank. převodem

		délka km	Kč/km	Kč	Kč celkem
Fakturujeme Vám za přepravu:	SPZ 1AK 6809				
17.5.	Ingolstadt				15 278,00
18.5.	Jažlovice				3 600,00
19.5.	Rehau				10 000,00
20.5.	Aš				3 340,00
21.5.	Libeznice				3 000,00

Celkem cena bez daně			**35 218,00**
Základ 1:	DS 1:	Daň 10%	0,00
Základ 2:	DS 2:	Daň 20%	7 043,60
Celkem cena s daní			42 261,60
K úhradě celkem	**Kč**		**42 262,00**

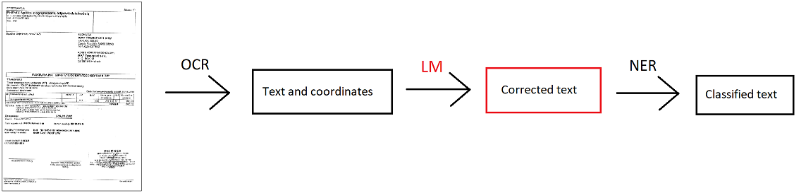
Razítko a podpis

```
confidence: 0.28953     prediction: 31001s,
confidence: 0.10326     prediction: aa*,
confidence: 0.04340     prediction: 2ia:',
confidence: 0.06670     prediction: &,
confidence: 0.16727     prediction: B*-,1,
```

Errors example

"S1."	"s1."
"21%"	"218"
"DPH"	"DPHT"
"ÚHRADĚ"	"ÚIIRADĚ"
"a"	"a"
"DIČ"	"DIč"
"www:"	"Wwww."

LM refinements



LM refinements

Using masked language modeling for error correction -> usage of broader context of the document.

- idea: low confidence ocr predictions are given to LM for correction

LM refinements

Using masked language modeling for error correction -> usage of broader context of the document.

- idea: low confidence ocr predictions are given to LM for correction
- Main goal: minimize word error rate

LM refinements

Using masked language modeling for error correction -> usage of broader context of the document.

- idea: low confidence ocr predictions are given to LM for correction
- Main goal: minimize word error rate
- Secondary goal is to minimize character error rate

LM refinements

Using masked language modeling for error correction -> usage of broader context of the document.

- idea: low confidence ocr predictions are given to LM for correction
- Main goal: minimize word error rate
- Secondary goal is to minimize character error rate
- Problem of tokenization, token vs character
    - elementary unit of ocr prediction: character
    - elementary unit of mlm prediction: token
    - mistake in a single character leads to change of several tokens

LM refinements

Input: context and word to be corrected (+ confidence)
Methods:

1. generative (purely contextual)
   - generate new prediction from scratch
   - : [T] [MASK] [T] -> [T] WO[MASK] [T] -> [T] WORD [T]
2. conservative
   - original word is taken into account
   - changing token one by one
   - possible usage of edit distance
   - : [T] [MASK]RO [T] -> [T] WO[MASK] [T] -> [T] WORD [T]

Generative

- works for short and one token words like prepositions, conjunctions and pronouns
- with more tokens gets very chaotic
- does not really work even when generating multiple candidates, most probably due to the structure of the document
- likely to damage correct ocr prediction

Chaos

```
desired output: a.s.            result: SlovenskoPočetkod.
desired output: Zakázka         result: C-symbol
desired output: Název           result: s.s
desired output: Faktura         result: .sp.
desired output: č.64            result: :--
desired output: s.r.o.          result: ]SK:SK:SK
desired output: a.s.            result: 2/191
desired output: Faktura         result: ##ka##žkaForma
desired output: vozovnou        result: /Praha-
desired output: DPH:            result: :::
desired output: r.o.            result: .,1/
desired output: IČO:            result: -SK:+
desired output: s.r.o.          result: Českárepublika,Českárepublika,
desired output: Slovenská       result: ČeskárepublikaČeská
desired output: DIČ:            result: ,00C
desired output: FAKTURA         result: '-'-'
desired output: Faktura         result: ##ka##žkaForma
desired output: dodávky         result: :Firma:
desired output: s.r.o.          result: .PrahaPrahaPrahaPrahaPraha
desired output: a.s.            result: protentonana
desired output: Fakturuji       result: (u##veden)
desired output: vnitrostátni            result: národníbanky,Brno,
desired output: MARVAL          result: 11.00
desired output: IČ:     result: -symbol:
desired output: a.s.            result: Slovensko,Slovensko,
desired output: DIČ:            result: -1111:
```

Conservative approach

There multiple ideas that can be implemented to conserve correct predictions.

- keep tokens that reach certain probability threshold according to the LM
- keep token if model is not sure about replacement
- use edit distance

Edit distance

For example, we accept change only if edit distance between new and original word is less than $x\%$ of length of the original word.

- works well for predictions with low character error rate
- can not repair very damaged words
- can be used as an criterion for selecting from the candidates
- also can help to reduce character error rate in general

Other possible upgrades

- iterate over the word multiple times (can be expensive)
- try multiple direction for correcting token (expensive)
- create multiple candidates for correction and then decide based on probability or/and edit distance (also possibly expensive)
- influence selection of the new token

Problems

- length (in tokens) of erroneous ocr prediction does not match the length of the desired output
  - how to find the resulting length
  - we can generate candidates of different length
  - function to compare candidates of different length
- more "features"means increased number of degrees of freedom, various thresholds
- efficiency / gain

Current results

repair ratio = $\frac{repaired}{total\_incorrect\_ocr\_pred}$, mistake ratio = $\frac{damaged}{total\_correct\_ocr\_pred}$,

- My simpler solution purely on probability and thresholds can reach repair ratio up to 30% with 10% mistake ratio. Incorrect correction tends to be worse than original incorrect prediction.
- The solution with additional edit distance threshold and two iteration correction can reach up to 36% repair ratio with about 4% mistake ratio. Additionally, incorrect correction does not increase character error rate.

Possible adaptations

- spell checker for identification of candidates
- instead of word, we receive probability distribution
- multiple view (most probably infeasible)

Sources

- https://github.com/JaidedAI/EasyOCR
- https://medium.com/doma/using-nlp-bert-to-improve-ocr-accuracy-385c98ae174c

MUNI
FAKULTA
INFORMATIKY