Two-step OCR Post-correction with BERT and Neural Machine Translation models

Master thesis

Author: Tanyu Tanev
Supervisor: Matthias Hertel
1st Examiner: Prof. Dr. Hannah Bast
2nd Examiner: Prof. Dr. Frank Hutter

Chair for Algorithms and Data Structures
Department of Computer Science
University of Freiburg

June 30, 2022

Table of Contents

Table of Contents

Tesseract OCR on Historical Document 1/2

ABSTRACT. It is shown that the assumption that language is non-finite involves the use of a constructive logic which leads to some restrictions on language theory and to the fact that the only possible definition of language is that proposed by generative grammars. Generative grammars can be formulated as normal /Markov/ algorithms and thus their study can be reduced to the study of such algorithms of a special type. A new type of generative grammar is defined, called matrix grammar. It is shown that a language generated by a context-restricted grammar can be also generated by a matrix grammar. Some properties of matrix grammars are shown to be decidable. The problem of the explicative power of generative grammars is discussed.

Figure: Excerpt from article [1] with its Abstract section

Tesseract OCR on Historical Document 2/2

ABSTRACT. It is shown that the assumption thrat language is non-finite involves the use of a constructive logic which leads to some restrictions on language theory and to the fact that the only rossitle definition of language is that proposec by generative gramrars. fGenerative grammars can be formulated asn normal /M¥arkov/ algorithms and thus their study can be reduced to the stufy of suck algorithms of a special +tyre. 4 new tyrpe of rsenerative grammar is defineé, called matrix grammar. It is shown that 2 languapge generated by a context-restricted grammar can be also generated by a matrix grammar. Some properties of matrix grammars are shown to be deecicable. The problem of the explicative power of generative granrmars is ciscussed.

Box: The resulting text reconstruction; red symbolizes mistakes

OCR Post-correction

- How can we fix erroneous OCR output...

OCR Post-correction

- How can we fix erroneous OCR output... with **OCR Post-correction**:

  "fGenerative grammars can be formulated asn normal /M¥arkov/ algorithms and thus their study can be reduced to the stufy of suck algorithms of a special +tyre."

  has to be repaired to

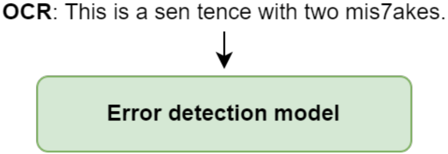
  "Generative grammars can be formulated as normal /Markov/ algorithms and thus their study can be reduced to the study of such algorithms of a special type."

Table of Contents

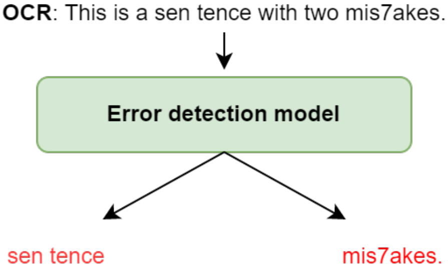
Two-step Approach

OCR: This is a sen tence with two mis7akes.

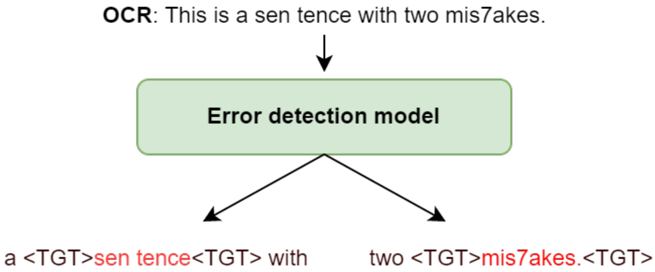
Two-step Approach



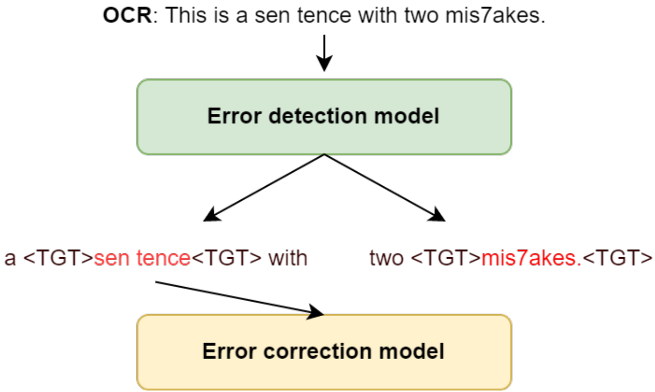
Two-step Approach



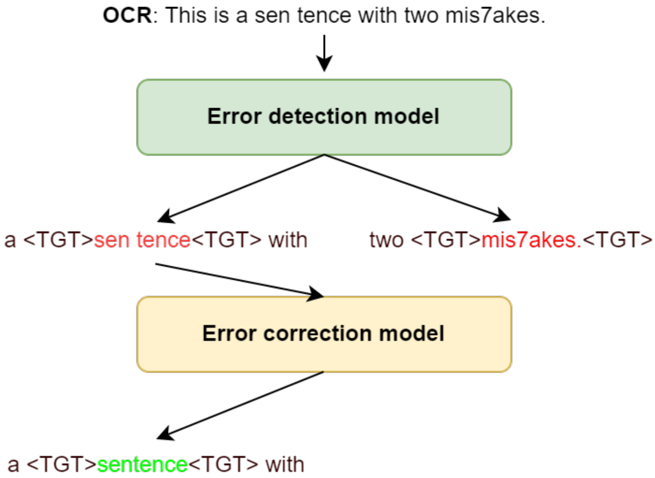
Two-step Approach



Two-step Approach



Two-step Approach



Two-step Approach

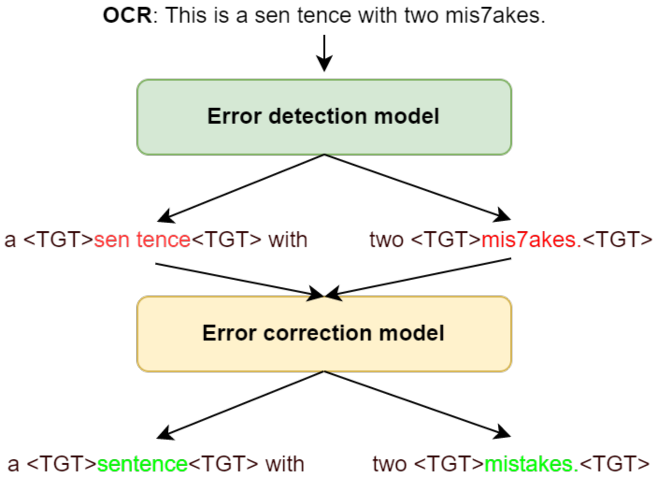


Figure: Visualization of **two-step** OCR Post-correction approach

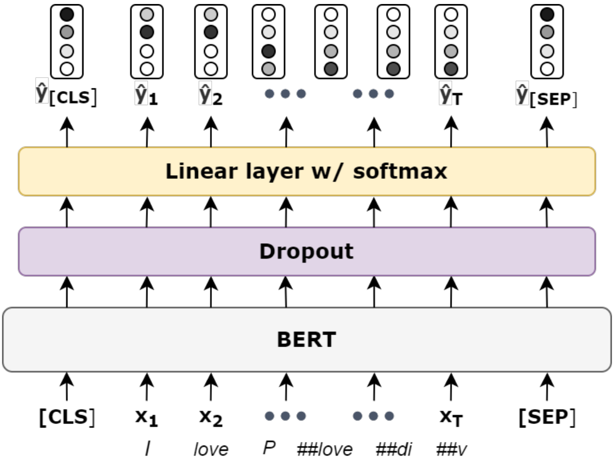
Error Detection 1/2

- **B**idirectional **E**ncoder **R**epresentations from **T**ransformers (or **BERT**):
  - *Pre-trained* on a **large** English dataset to "understand" language
  - **Fine-tuned** for downstream task (i.e., OCR error *detection*)
  - Uses a **subword** tokenizer:

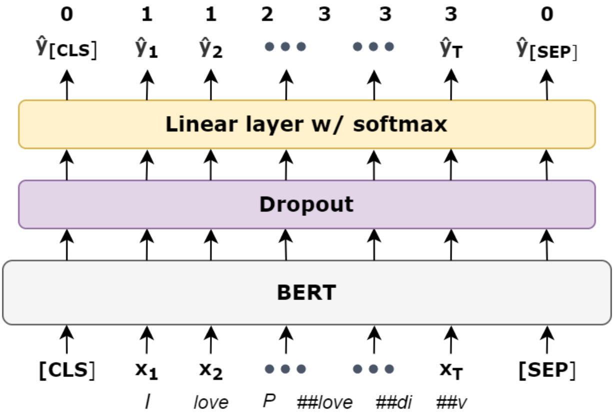
Plovdiv → ['P', '##lov', '##di', '##v']
*Plov**e**div* → ['P', '##lov**e**', '##di', '##v']

- Middle ground between **character** and **word** tokenization
- Flexibility of character tokenization (no OOV errors) ✓
- Power of word tokenization (more context than just chars) ✓

Error Detection 2/2



Error Detection 2/2



Error Detection 2/2

1	**1**	**2**	**3**	**3**	**3**
I	love	P	##love	##di	##v

Error Detection 2/2

1	**1**	**2**	**3**	**3**	**3**
I	love	<TGT>P	##love	##di	##v<TGT>

Error Detection 2/2

I love <TGT>Plovediv<TGT>

Figure: Visualization of using a BERT model for OCR **error detection**

Error Correction 1/2

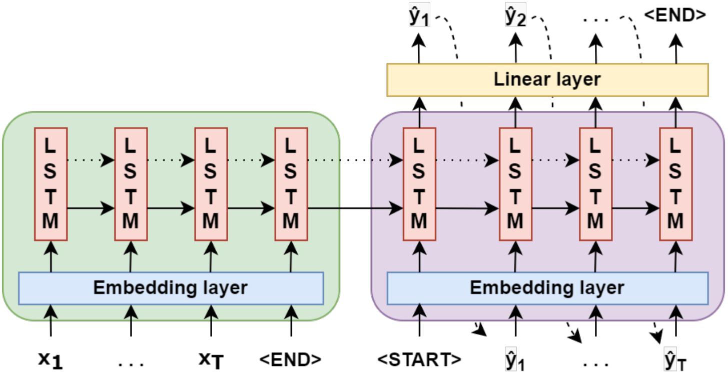


Figure: Workflow for LSTM sequence-to-sequence error correction model

Error Correction 2/2

- Will be evaluating two models:
  - LSTM sequence-to-sequence w/ and w/o **attention**
  - Transformer: does character-level **attention** work well?

Error Correction 2/2

- Will be evaluating two models:
  - LSTM sequence-to-sequence w/ and w/o **attention**
  - Transformer: does character-level **attention** work well?
- **NB 1:** Preceding and succeeding contexts can *also* contain errors
  $\rightarrow$ *multiple* correction samples with **one** target token each
- **NB 2:** No *error-free* samples are used for training

Table of Contents

Detection Metrics

- How to classify the **token** predictions of the detection model?

Detection Metrics

- How to classify the **token** predictions of the detection model?
  - Token was erroneous, and model found it ⇒ **true positive**
  - Token was *not* erroneous, but model found it as such ⇒ **false positive**
  - Token was erroneous, but model *did not* find it ⇒ **false negative**

Detection Metrics

- How to classify the **token** predictions of the detection model?
  - Token was erroneous, and model found it $\Rightarrow$ **true positive**
  - Token was *not* erroneous, but model found it as such $\Rightarrow$ **false positive**
  - Token was erroneous, but model *did not* find it $\Rightarrow$ **false negative**
- Standard **information retrieval** metrics:
  - **Precision**: $\frac{TP}{TP+FP}$
    $\rightarrow$ how many error predictions were **actually** errors
  - **Recall**: $\frac{TP}{TP+FN}$
    $\rightarrow$ how many of the **expected** errors were predicted
  - **F1 score**: $2 * \frac{precision * recall}{precision + recall}$
    $\rightarrow$ **harmonic mean** of the *recall* and *precision*

Correction Metrics 1/2

- **IDEA:** BERT will take care of *marking* the erroneous tokens; the correction models need to be able to **correct** them properly

$$\textit{love}\ \underbrace{<\textbf{TGT}>\textbf{Plovediv}<\textbf{TGT}>}_{\text{Target token}}$$

- Metric: % change of **Levenshtein distance** between **target tokens**

Input: *love <TGT>Plov**e**div<TGT>*
Prediction: *lovd <TGT>Plovdiv<TGT>*
Target: *love <TGT>Plovdiv<TGT>*

Correction Metrics 2/2

- How to measure correction performance of *full pipeline*?
- **IDEA:** Measure impact of using two-step model on all texts
  $\rightarrow$ did it *help* or make things *worse*?
- **How?**
  - Calculate *sum* of Levenshtein distances in all **original texts**
  - Calculate *sum* of Levenshtein distance in all **predicted texts**
  - Determine the % change between the two sums

Datasets

- ICDAR2017* — pre 19th century literature and publications
    - Monograph (e.g., books)
    - Periodical (e.g., newspapers, magazines)
- ICDAR2019* — **highly erroneous** old literature
- "Pure OCR Errors"* — collection of *automatically* extracted OCR errors from the ACL Anthology Reference Corpus
- "ACL Benchmark" — randomly sampled and **manually corrected** OCR errors from the ACL Anthology Reference Corpus
- Artificial data* — **error statistics** + clean dataset

* - used for training

Comparison bases

- Baseline **dictionary** approach:
  - If word not in dictionary → it's erroneous
  - Corrections w/ **Q-gram index**: *lowest* ED and *highest* freq.
- External models:
  - NATAS [2] — **character-level** NMT model with *vanilla* RNN cells and *general* Luong attention [3]
  - Google Autocorrect —— random subset of 100 samples; accept corrections until none are left
- Competition models:
  - Char-SMT/NMT [4] - hybrid model w/ **NMT** for detection and **SMT** for correction
  - WFST-PostOCR - vocabulary + weighted finite-state transducers
  - CCC - *multilingual* **BERT** for detection + **LSTM encoder-decoder** for correction
  - Nguyen et al. - **BERT** for detection + **LSTM encoder-decoder** for correction (simplified when compared to CCC)

Final Detection Results

	ACL	ICDAR2017 monograph	ICDAR2017 periodical	ICDAR2019
Plain Training Q-Index w/ max. dist. 3	28.53%	49.2%	35.04%	44.71%
Char-SMT/NMT	x	67%	64%	x
WFST-PostOCR	x	73%	68%	x
CCC	x	x	x	67%
Nguyen et al.	x	72%	74%	68%
Google Autocorrect	36.93%			
NATAS	10.05%	27.53%	23.54%	28.42%
Big Unfrozen BERT	51.61%	57.13%	52.37%	42%

Table: Subset of final results for OCR error detection on the **testing** datasets w/ **F1 score**

Final Correction Results

	ACL	ICDAR2017 monograph	ICDAR2017 periodical	ICDAR2019
Plain Training Q-Index w/ max. dist. 3	-76.52%	-52.1%	-52.33%	-46.28%
Char-SMT/NMT	x	+43%	+37%	x
WFST-PostOCR	x	+28%	x	x
CCC	x	x	x	+11%
Nguyen et al.	x	+36%	+27%	+4%
Google Autocorrect	-21%			
NATAS	-92.5%	-81.84%	-81.16%	-75%
2-step w/ (3,3) LSTM	-8.1%	+2.38%	-7.3%	-9.72%
2-step w/ isolated Transf.	-8.2%	+2.2%	-9.99%	-10.67%

Table: Final results on the **testing** datasets w/ best-performing models, given in **% improvement of the sum of Levenshtein distances**

Distribution of Detector-Generated Samples

	ACL	ICDAR2017 monograph	ICDAR2017 periodical	ICDAR2019
Plain Training Q-Index	17%/77%	38%/67%	26%/43%	36%/50%
Google Autocorrect	49%/29%			
NATAS	5%/52%	17%/64%	14%/49%	17%/52%
BERT + (3,3)	56%/37%	65%/47%	57%/39%	48%/22%
BERT + Isolated	57%/37%	67%/47%	58%/39%	50%/22%

Table: Metrics measuring how many of the detector-generated samples are missed/superfluous; first percent is **precision**, the second is **recall**

Final Correction Results on Matches

	ACL	ICDAR2017 monograph	ICDAR2017 periodical	ICDAR2019
Plain Training Q-Index w/ max. dist. 3	+9.33%	+19.78%	-1.7%	-8.45%
Google Autocorrect	+23.88%			
NATAS	-46.67%	-26.73%	-44.76%	-43.43%
(3,3) LSTM	+35.57%	+52.72%	+41.97%	+34.17%
Isolated Transf.	+43.92%	+53.29%	+37.17%	+30%

Table: Performance of a subset of the different models from the paper exclusively on the group of **correctly matched** correction samples

Conclusion

- Transformer models are competitive with LSTM w/ attention
  ... but require a lot more data to train with *context*
- Error **detection** is the *bottleneck* of a two-step approach
  $\rightarrow$ Remedy idea: expose correction model to **error-free** data, and focus on high detection **recall**
- It is **very difficult** to create a generic OCR correction model, which works well across *all domains*
  $\rightarrow$ Promising research direction: artificial generation of **domain-specific** data

Thank you for your time and attention!

References I

S. Abraham, "Some questions of language theory," in *COLING 1965*, 1965.

M. Hämäläinen and S. Hengchen, "From the paft to the fiiture: a fully automatic NMT and word embeddings method for OCR post-correction," *CoRR*, vol. abs/1910.05535, 2019.

M. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *CoRR*, vol. abs/1508.04025, 2015.

C. Amrhein and S. Clematide, "Supervised ocr error detection and correction using statistical and neural machine translation methods," *Journal for Language Technology and Computational Linguistics (JLCL)*, vol. 33, no. 1, pp. 49–76, 2018.

References II

P. Estrella and P. Paliza, "Ocr correction of documents generated during argentina's national reorganization process," in *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, DATeCH '14, (New York, NY, USA), p. 119–123, Association for Computing Machinery, 2014.

R. Schaefer and C. Neudecker, "A two-step approach for automatic ocr post-correction," in *LATECHCLFL*, 2020.

T. T. H. Nguyen, A. Jatowt, N.-V. Nguyen, M. Coustaty, and A. Doucet, *Neural Machine Translation with BERT for Post-OCR Error Detection and Correction*, p. 333–336.
New York, NY, USA: Association for Computing Machinery, 2020.

References III

J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018.

Appendix TOC

Table of Contents

Dictionary Approach 1/5

- One classical OCR Post-correction approach is using a **dictionary** [5]:
  - Accumulate a **large** collection (i.e., dictionary) of *valid* words
  - Split OCR-ed texts by *whitespace*
  - Check each word against the dictionary:
    - If it is **known** (does not equal **correct**), leave it
    - If it is **not known**, propose a *correction*

Dictionary Approach 2/5

- How to propose good corrections...

[1] The most common set of edit operations is also called **Levenshtein operations**

Dictionary Approach 2/5

- How to propose good corrections. . . by using **Edit Distance** (or **ED**):
  - Determine the **minimal** number of **single-character** operations, in order to *transform* one string into another

[1] The most common set of edit operations is also called **Levenshtein operations**

Dictionary Approach 2/5

- How to propose good corrections. . . by using **Edit Distance** (or **ED**):
  - Determine the **minimal** number of **single-character** operations, in order to *transform* one string into another
  - Permissible operations (in most common case [1]):
    - **Insertion**: *Plovdiv* → *Plovediv*

[1] The most common set of edit operations is also called **Levenshtein operations**

Dictionary Approach 2/5

- How to propose good corrections... by using **Edit Distance** (or **ED**):
  - Determine the **minimal** number of **single-character** operations, in order to *transform* one string into another
  - Permissible operations (in most common case [1]):
    - **Insertion**: *Plovdiv* → *Plovediv*
    - **Deletion**: *Plovdiv* → *Plodiv*

[1] The most common set of edit operations is also called **Levenshtein operations**

Dictionary Approach 2/5

- How to propose good corrections... by using **Edit Distance** (or **ED**):
  - Determine the **minimal** number of **single-character** operations, in order to *transform* one string into another
  - Permissible operations (in most common case [1]):
    - **Insertion**: *Plovdiv* $\rightarrow$ *Plovediv*
    - **Deletion**: *Plovdiv* $\rightarrow$ *Plodiv*
    - **Substitution**: *Plovdiv* $\rightarrow$ *Plofdiv*

[1] The most common set of edit operations is also called **Levenshtein operations**

Dictionary Approach 2/5

- How to propose good corrections... by using **Edit Distance** (or **ED**):
  - Determine the **minimal** number of **single-character** operations, in order to *transform* one string into another
  - Permissible operations (in most common case [1]):
    - **Insertion**: *Plovdiv* → *Plovediv*
    - **Deletion**: *Plovdiv* → *Plodiv*
    - **Substitution**: *Plovdiv* → *Plofdiv*
  - *Low* edit distance → **similar**; *high* edit distance → **different**
  - Tie-breaker: word **frequency** (i.e., how often was word *encountered* when accumulating words for dictionary)

[1] The most common set of edit operations is also called **Levenshtein operations**

Dictionary Approach 3/5

Vocabulary: [('this', 4), ('is', 4), ('a', 2), ('cat', 2), ('rad', 1), ('bad', 1)]

OCR: *"This is a rat."*

Correction candidates:

rat → this (ED 4)
rat → is (ED 3)
rat → a (ED 2)
rat → cat (ED 1, frequency 2)
rat → rad (ED 1, frequency 1)
rat → bad (ED 2)

Dictionary Approach 4/5

- **Problem**: Comparing each word against a *large* word collection is **expensive**
- **Solution**:

Dictionary Approach 4/5

- **Problem**: Comparing each word against a *large* word collection is **expensive**
- **Solution**: **Q-grams**
  - A Q-gram is a *substring* of length q
  - Similar words (i.e., with **low ED**) *must* have many **common** substrings
    $\rightarrow$ the other ones (w/ few shared substrings) can be **skipped**
  - Practical threshold: $comm(x, y) \geq max(|x|, |y|) - 1 - (\delta - 1) * q$, with:
    - $comm(x, y)$: # shared Q-grams between strings x and y
    - $|x|$: length of arbitrary string x
    - δ: maximum allowed edit distance
    - q: the size (i.e., length) of the Q-grams

Dictionary Approach 5/5

- Shortcomings of baseline approach:
  - **Real-word** errors: valid words that do not fit *in context* (see last slide)
  - **Named entities**: names and acronyms are **not** valid words
  - **Word boundary** errors: addition/deletion of **whitespaces**
    - **Run-on** error: *In Plovdiv* $\rightarrow$ *InPlovdiv*
    - **Incorrect split** error: *Plovdiv* $\rightarrow$ *Pl ovdiv*

Table of Contents

Two-step Approach Motivation

- Recent research (see [6] and [7]) has started using two **separate** deep-learning models for error *detection*, and then *correction*
- **Motivation**:

Two-step Approach Motivation

- Recent research (see [6] and [7]) has started using two **separate** deep-learning models for error *detection*, and then *correction*
- **Motivation**:
  - Allows usage of powerful detection model - **BERT** [8]

Two-step Approach Motivation

- Recent research (see [6] and [7]) has started using two **separate** deep-learning models for error *detection*, and then *correction*
- **Motivation**:
  - Allows usage of powerful detection model - **BERT** [8]
  - Reduces the amount of "overcorrected" samples (e.g., *"This is my car"* to *"This is my cat"*)

Two-step Approach Motivation

- Recent research (see [6] and [7]) has started using two **separate** deep-learning models for error *detection*, and then *correction*
- **Motivation**:
  - Allows usage of powerful detection model - **BERT** [8]
  - Reduces the amount of "overcorrected" samples (e.g., *"This is my car"* to *"This is my cat"*)
  - Allowing the error correction model to focus on that task only *should* theoretically boost its performance [6]

Two-step Approach Evaluation

- Group **detection-generated** and **expected** correction samples in:
  - **Matched**: Detection-generated $\cap$ Expected
  - **Missed**: Expected $\setminus$ Detection-generated
  - **Superfluous**: Detection-generated $\setminus$ Expected

Two-step Approach Evaluation

- Group **detection-generated** and **expected** correction samples in:
  - **Matched**: Detection-generated $\cap$ Expected
  - **Missed**: Expected $\setminus$ Detection-generated
  - **Superfluous**: Detection-generated $\setminus$ Expected
- Then, *% change of Levenshtein distance sum* on:

Two-step Approach Evaluation

- Group **detection-generated** and **expected** correction samples in:
  - **Matched**: Detection-generated $\cap$ Expected
  - **Missed**: Expected $\setminus$ Detection-generated
  - **Superfluous**: Detection-generated $\setminus$ Expected
- Then, *% change of Levenshtein distance sum* on:
  - **Original** texts: sum of *missed* and *matched* samples

Two-step Approach Evaluation

- Group **detection-generated** and **expected** correction samples in:
  - **Matched**: Detection-generated $\cap$ Expected
  - **Missed**: Expected $\setminus$ Detection-generated
  - **Superfluous**: Detection-generated $\setminus$ Expected
- Then, *% change of Levenshtein distance sum* on:
  - **Original** texts: sum of *missed* and *matched* samples
  - **Predicted** texts: sum of *missed*, *matched* **and** superfluous samples

Table of Contents

Error Type Statistics

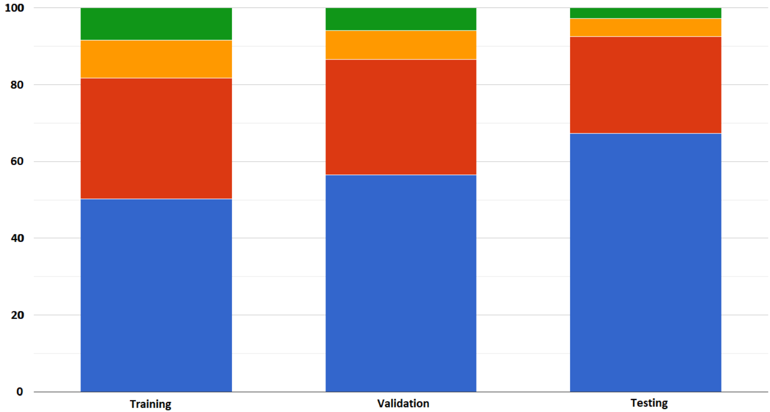


Figure: blue represents **single**-mistake errors; red represents **double**-mistake errors; yellow represents **triple**-mistake errors; green represents **multi**-mistake errors

Word Boundary Error Statistics

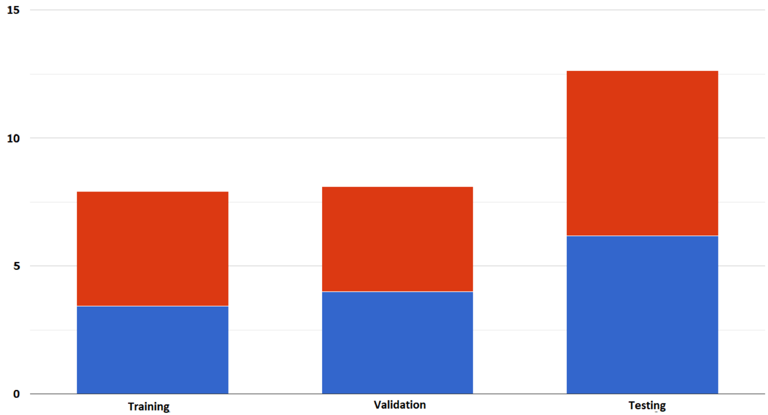


Figure: blue represents **run-on** errors; red represents **incorrect split** errors

Table of Contents

Artificial Data 1/2

- **How?** Record statistics:
  - How often are letters substituted with other *combinations* (1 or 2 symbols)?
  - What combinations of *edit operations* are typical?
    e.g., del→sub is most often a **double-character** substitution (*/) to *p*)
  - At which **positions** do these edit operations happen?
    → not randomly distributed (example again: double-character substitutions)
- **From where?** ICDAR datasets + "Pure OCR Errors"
- Generate until a custom set *threshold* is hit (based on number of words already handled)

Artificial Data 2/2

- For training the final models: **200,000-word limit**
- Edit operation statistics:
  - 3.53% insertions
  - 7% deletions
  - 17.82% substitutions
- Error type statistics:
  - 65.44% single-mistake
  - 26.42% double-mistake
  - 5.33% triple-mistake
  - 2.81% multi-mistakes (i.e., four mistakes or more)
- Word boundary error statistics:
  - 3.1% run-on
  - 4.45% incorrect split

Table of Contents

Q-Index Experiment Results 1/2

		ACL	ICDAR2017 monograph	ICDAR2017 periodical	ICDAR2019
Training Q-index	Plain	28.53%	49.2%	35.04%	44.71%
	Skip NE	27.93%	48.13%	35.36%	43.26%
ArXiv Q-index	Plain	34.49%	46.58%	32.23%	42.14%
	Skip NE	30.17%	47.71%	35.1%	43.45%

Table: Error detection results for different experiments with baseline Q-index model, evaluated on the **test** datasets

Q-Index Experiment Results 2/2

		ACL	ICDAR2017 monograph	ICDAR2017 periodical	ICDAR2019
Training Q-index	Plain	+7.8%	+16.33%	+1.62%	-4.46%
	Skip NE	+7.34%	+14.42%	-1.56%	-11.92%
ArXiv Q-index	Plain	+9.48%	+14.26%	-1.88%	-11.99%
	Skip NE	+8.26%	+12.75%	-4.38%	-14.56%

Table: Error correction results for different experiments with baseline Q-index model, evaluated on the **test** datasets

Table of Contents

Mixed Dataset Experiment Results

	ACL	ICDAR2017 monograph	ICDAR2017 periodical	ICDAR2019	Pure OCR Errors*
Only ICDAR2017 monograph (4:15:30)	-40.09%	+34.21%			
Both ICDAR2017 datasets (4:29:38)	-32.43%	+32.69%	+25.46%		
Both ICDAR2017 + ICDAR2019 (4:28:50)	-28.76%	+33.37%	+23.97%	+5.09%	
All ICDAR + Pure OCR Errors (4:59:52)	-14.64%	+33.64%	+25.56%	+5.64%	+32.18%
All ICDAR + Pure OCR Errors + Artificial 200k (6:09:54)	+3.79%	+31.11%	+24.02%	+4.27%	+41.99%

Table: Results for running a *(3,3)* context Transformer model on different "mixes" of datasets, evaluated on the **validation** datasets

Table of Contents

LSTM Experiments 1/2

		ACL	ICDAR2017 monograph	ICDAR2017 periodical	ICDAR2019	Pure OCR Errors*
Context size	1, 1 (4:01:43)	-9.66% (4% missed)	+37.23% (3% missed)	+31.65% (2% missed)	+18% (2% missed)	+39.47% (3% missed)
	3, 3 (5:20:00)	-7.96% (3% missed)	+32.87% (4% missed)	+30.84% (2% missed)	+19.65% (3% missed)	+36.04% (5% missed)
	5, 5 (8:39:29)	-8.52% (2% missed)	+36.43% (2% missed)	+29.27% (1% missed)	+21.62% (0.9% missed)	+37.66% (2% missed)
	5, 1 (5:20:48)	-7.17% (2% missed)	+35.75% (3% missed)	+31.48% (2% missed)	+20.53% (2% missed)	+40.96% (2% missed)

Table: First subset of results for different experiments with an LSTM encoder-decoder correction model, evaluated on the **validation** datasets

LSTM Experiments 2/2

		ACL	ICDAR2017 monograph	ICDAR2017 periodical	ICDAR2019	Pure OCR Errors*
Attention Type	None (5:20:02)	-34.45% (1% missed)	+14.88% (2% missed)	+0.63% (1% missed)	-7.16% (2% missed)	+24.78% (2% missed)
	Dot (5:42:16)	-6.48% (2% missed)	+32.53% (2% missed)	+30.22% (1% missed)	+20.12% (2% missed)	+36.82% (2% missed)
	General (5:36:18)	-8.41% (0.8% missed)	+35.92% (2% missed)	+31.83% (0.6% missed)	+17.51% (1% missed)	+39.27% (2% missed)
	Concat (8:08:41)	-5.14% (0.3% missed)	+35.64% (1% missed)	+30.98% (0,6% missed)	+16.23% (0.5% missed)	+39% (0.75% missed)

Table: Second subset of results for different experiments with an LSTM encoder-decoder correction model, evaluated on the **validation** datasets

Transformer Experiments

		ACL	ICDAR2017 monograph	ICDAR2017 periodical	ICDAR2019	Pure OCR Errors*
Context size	1, 1 (5:25:38)	-14.98%	+39.29%	+27.4%	+16.44%	+41.37%
	3, 3 (8:35:54)	-15.19%	+39.27%	+27.91%	+13.46%	+39.23%
	5, 5 (13:37:10)	-18.33%	+37.31%	+25.84%	+13.68%	+34.23%
	5, 1 (8:30:33)	-16.35%	+37.19%	+27.22%	+14.18%	+38.93%

Table: Subset of results for different experiments with a Transformer correction model, evaluated on the **validation** datasets

BERT Detection Experiments

		ACL	ICDAR2017 monograph	ICDAR2017 periodical	ICDAR2019	Pure OCR Errors*
Fine tuning	Unfr. emb. + no fr. BERT layers (8:46:42)	60.87%	68.64%	63.51%	65.34%	88.91%
	Unfr. emb. + fr. nine layers (7:16:08)	57.24%	67%	62.03%	60.93%	90.17%
	Fr. emb. + fr. nine layers (5:39:42)	58.25%	66.67%	62.46%	60.54%	89.13%
	Fr. emb. + fr. all layers (4:44:51)	34.1%	42.75%	41.14%	33.19%	58.37%

Table: Subset of results for different experiments with a BERT detection model, evaluated on the **validation** datasets with classification threshold *0.98*

Table of Contents

Full Final Correction Results 1/2

	ACL	ICDAR2017 monograph	ICDAR2017 periodical	ICDAR2019
Plain Training Q-Index w/ max. dist. 3	-76.52%	-52.1%	-52.33%	-46.28%
Char-SMT/NMT	x	+43%	+37%	x
WFST-PostOCR	x	+28%	x	x
CCC	x	x	x	+11%
Nguyen et al.	x	+36%	+27%	+4%
Google Autocorrect	-21%			
NATAS	-92.5%	-81.84%	-81.16%	-75%
2-step w/ (3,3) LSTM	-8.1%	+2.38%	-7.3%	-9.72%
2-step w/ isolated LSTM	-8.45%	+1.56%	-9.31%	-10.61%
2-step w/ (3,3) Transf.	-12.99%	-4.3%	-11.95%	-14.31%
2-step w/ isolated Transf.	-8.2%	+2.2%	-9.99%	-10.67%

Table: Final results on the **testing** datasets, given in **% improvement of the sum of Levenshtein distances**

Full Final Correction Results 2/2

	ACL	**ICDAR2017 monograph**	**ICDAR2017 periodical**	**ICDAR2019**
Plain Training Q-Index w/ max. dist. 3	+9.33%	+19.78%	-1.7%	-8.45%
Google Autocorrect	+23.88%			
NATAS	-46.67%	-26.73%	-44.76%	-43.43%
(3,3) LSTM	+35.57%	+52.72%	+41.97%	+34.17%
Isolated LSTM	+37.65%	+49.3%	+33.72%	+26%
(3,3) Transf.	+31.23%	+52.76%	+38.54%	+26.25%
Isolated Transf.	+43.92%	+53.29%	+37.17%	+30%

Table: Performance of the different models from the paper exclusively on the group of **correctly matched** correction samples

Detection Group Results

		ACL	ICDAR2017 monograph	ICDAR2017 periodical	ICDAR2019
Plain Training Q-Index w/ max. dist. 3	Matched	16.45%	31.62%	19.45%	26.34%
	Missed	4.82%	15.86%	25.64%	25.86%
	Superfluous	78.73%	52.52%	55%	47.8%
Google Autocorrect	Matched	22.44%			
	Missed	54.15%			
	Superfluous	23.41%			
NATAS	Matched	5.2%	15.33%	11.92%	14.91%
	Missed	4.77%	8.92%	12.36%	13.82%
	Superfluous	90%	75.92%	75.72%	71.25%
BERT + (3,3)	Matched	28.63%	37.8%	30%	17.47%
	Missed	48.66%	42.1%	47.47%	63.53%
	Superfluous	22.71%	20.1%	22.58%	19%
BERT + Isolated	Matched	29.01%	38.51%	30.42%	18.29%
	Missed	48.66%	42.83%	47.38%	63.42%
	Superfluous	22.33%	18.66%	22.2%	18.29%

Table: Fractions of groups of **detection-generated** samples

Table of Contents

Detection sample group influence

- Given: 2-step **isolated** Transformer model on **ICDAR2017**

Detection sample group influence

- Given: 2-step **isolated** Transformer model on **ICDAR2017**
- Sum of Levenshtein distances of:
  - **Matched** samples: 7,985 $\rightarrow$ 3,730 (**+53.29%**)

Detection sample group influence

- Given: 2-step **isolated** Transformer model on **ICDAR2017**
- Sum of Levenshtein distances of:
  - **Matched** samples: 7,985 $\rightarrow$ 3,730 (**+53.29%**)
  - **Superfluous** samples: **3,884**

Detection sample group influence

- Given: 2-step **isolated** Transformer model on **ICDAR2017**
- Sum of Levenshtein distances of:
  - **Matched** samples: 7,985 $\rightarrow$ 3,730 (**+53.29%**)
  - **Superfluous** samples: **3,884**
  - **Missed** samples: **8,858**

Worse “ACL Benchmark” Performance 1/4

- Mismatched/hard cases:
  - *“of* ***<TGT>GmeptWizatiofls<TGT>*** *Urderlying”* → *“of* ***<TGT>Conceptualizatio@ns<TGT>*** *Urderlying”*
  - *“be* ***<TGT>awit r∼_mmchaw<TGT>****”* → *“be* ***<TGT>switched somehow.<TGT>****”*
  - *“or* ***<TGT>∼ixsemglL∼.<TGT>****”* → *“or* ***<TGT>polysem@ous.<TGT>****”*
  - *“of* ***<TGT>’IIYXEght<TGT> and****”* → *“of* ***<TGT>@@Thought<TGT> and****”*

Worse "ACL Benchmark" Performance 2/4

- Named entities:
  - *"Francism, <**TGT**>**Qlifomia.**<**TGT**>"* → *"Francism, <**TGT**>**California.**<**TGT**>"*
  - *"4-3-11 <**TGT**>**T∼keda,**<**TGT**> Kofu"* → *"4-3-11 <**TGT**>**Takeda,**<**TGT**> Kofu"*

Worse “ACL Benchmark” Performance 3/4

- Formulas/Technical jargon:
  - “i: <TGT>f∼(X,y)<TGT> =” → “i: <TGT>fi(X,Yy)<TGT> =”
  - “by: <TGT>(T∼)-1,<TGT> if” → “by: <TGT>(Tiˆm)ˆ-1,<TGT> if”
  - “<TGT>=∼<TGT> ('–*OR” → “<TGT>=><TGT> ('–*OR”

Worse “ACL Benchmark” Performance 4/4

- Non-English sequences (specifically: German)
  - *“**<TGT>P∼dagogischer<TGT>** Verlag”* → *“**<TGT>Pädagogischer<TGT>** Verlag”*
  - *“einem **<TGT>Gener\]erungssystem<TGT>** fHr”* → *“einem **<TGT>Gener@ierungssystem<TGT>** fHr”*

Missed samples 1/8

- **Proper misses**:
  - *multiword entl.t∼es appear*
  - *referential miscommunicatiou, having*
  - *elements tha.t also*
  - *and (senti-) automatic*
  - *Amidst the arte which*
  - *his colleigues.*

Missed samples 2/8

- **Punctuation mistakes**:
  - ',' to '.' for sentence end
  - '.' to ',' for sentence continuation
  - '?' to apostrophes:
    - *<TGT>Teachers?<TGT> Estimates* → *<TGT>Teachers'<TGT> Estimates*
    - *<TGT>?innate ?language<TGT>* → *<TGT>“innate” language<TGT>*
  - Fixing citations:
    - *<TGT>\ [Robinson,<TGT> <TGT>1982\].<TGT>* → *<TGT>@[Robinson,<TGT> <TGT>1982@].<TGT>*
    - *<TGT>\[Church<TGT> et aL, <TGT>1991\]<TGT>* → *<TGT>@[Church<TGT> et aL, <TGT>1991@]<TGT>*
  - “Fixing” sequences:
    Huang. XD. Hen. HW. and Lee. KP.. → *Huang., XD., Hen., HW., and Lee. KP..*

Missed samples 3/8

- **Jargon**:
  - **Formulas**:
    - *"s(iek)"* $\rightarrow$ *"s(i,k)"*
    - *"c(t)"* $\rightarrow$ *"c(i)"*
  - **Non-English**:
    - *Er gibt mir Wein Er <TGT>stelgt<TGT> mir auf <TGT>den'<TGT> <TGT>Fu/3<TGT>*
    - *Sofia <TGT>∼niversitat<TGT> Heidelberg-Konstanz*

Missed samples 4/8

- **Named entities**:
  - *<TGT>North-Ilolland,<TGT> Amsterdam*
  - *Stanford, <TGT>Callfo∼n\[a<TGT>*
  - *<TGT>(infcrcncc)<TGT> <TGT>hdy<TGT> might* $\rightarrow$ *<TGT>(inference)<TGT> <TGT>Andy<TGT> might*
  - *WOOLLEN MANUFACTURERS, <TGT>W0LSI5GIMM.<TGT>* $\rightarrow$ *WOOLLEN MANUFACTURERS, <TGT>WOLSINGHAM.<TGT>*
  - *Mr. <TGT>Fusler<TGT> with an interview* $\rightarrow$ *Mr. <TGT>Fowler<TGT> with an interview*
  - *Rev. T. <TGT>Shmelev<TGT> treasurer* $\rightarrow$ *Rev. T. <TGT>Stomeley<TGT> treasurer*

Missed samples 5/8

- **Incorrectly marked boundaries**:
  - *<TGT>P.tl.<TGT>* → *P.<TGT>tl.<TGT>* (*<TGT>P.@H.<TGT>*)
  - *<TGT>recognitmn".<TGT>* → *<TGT>recognitmn<TGT>".* (*<TGT>recognition".<TGT>*)
  - *<TGT>svlected.<TGT>* → *<TGT>svlected<TGT>.* (*<TGT>selected,<TGT>*)
  - *The <TGT>DIONR<TGT> <TGT>.s.<TGT> or other steamer* → *The <TGT>DIONR .s.<TGT> or other steamer* (*The <TGT>DIONE<TGT> <TGT>s.s.<TGT> or other steamer*)

Missed samples 6/8

- **Hard cases**:
  - *We have not yet examined in full <TGT>thoee<TGT> cases where de-elefting leaves a state-expression.*
  - *IIL LEARNING AND RECOGNITION <TGT>PIIASES<TGT>*
  - *of his Third 'Elements <TGT>hy<TGT> which he*
  - *<TGT>124118<TGT> S-. (<TGT>124 l. 18<TGT> S-.)*
  - *[A ? ? ? a?, i, j](z1, x1) : [A ? ?a ? ?, i, j + 1] ???(y1 : A ? ?a?) ? P 0 ? i ?*
  - *<TGT>i tasehold<TGT> (<TGT>@Leasehold<TGT>)*

Missed samples 7/8

- **Inexplicable addition**:
  - *<TGT>46<TGT> ROBERT THE DEtJTLL. → <TGT> 46<TGT> ROBERT THE DEtJTLL.*
  - *Retrieval <TGT>3000<TGT> documents → Retrieval <TGT>∼3000<TGT> documents*
  - *Hindoo and Muiiumedan <TGT>Period<TGT> → Hindoo and Muiiumedan <TGT>Periods.<TGT>*

Missed samples 8/8

- **Incorrect "corrections"** (mainly ICDAR2019):
  - *<TGT>considered,<TGT>* → *<TGT>conffder'd,<TGT>*
  - *for the most <TGT>part<TGT>* → *for the most <TGT>pare<TGT>*
  - *<TGT>first<TGT>* → *<TGT>@first<TGT>*
  - *<TGT>offering<TGT>* → *<TGT>o@ffering<TGT>*

Superfluous samples 1/4

- **Missed errors in dataset**:
  - *"multiple cycles of protottyplng."*
  - *The basic idea. . . i very simple*
  - *depends on its?o,o head in the relation*
  - *taken no fee strictly oonbdenl iol .lisbuiee no <TGT>obleet<TGT>*
  - *een pleased to appoint*
  - *their theological dif-ference ,*
  - *tmlsome and adventurous these expeditions*
  - *<TGT>Sureeon-Dentist,<TGT>*
  - *instantly curing tooth-ache, atiu rendering*
  - *hereby giv notice,*

Superfluous samples 2/4

- **Looks** like it should be an error:
  - *other <TGT>pe-souul<TGT> Estate*
  - *A <TGT>enm fortable<TGT> smoke-room,*
  - *from such announcement, but <TGT>r .,<TGT> assume*
  - *had not been p. anu kn <TGT>Deoeased<TGT>*
  - *desirable residences for <TGT>gei<TGT> families*
  - *the aged Mr. <TGT>B- conduet<TGT> his family worship,*
  - *comes the sad <TGT>oHmax-when<TGT> Durham,*

Superfluous samples 3/4

- **Punctuation**:
  - *the same is <TGT>sum<TGT> moned as much*
  - *spice broths sre too <TGT>hot-Treason's<TGT> in a December*
  - *demise of Lord <TGT>Bruc t -<TGT> the t son*
  - *to warm their <TGT>sit ting<TGT> rooms.*
  - *of her <TGT>Majesty's<TGT> Treasury,*
  - *<TGT>-London<TGT>, 22, Pall-mall.*
  - *<TGT>How-ever,<TGT> for the matter of vanity,*
  - *<TGT>hav-ing<TGT> known him from youth*

Superfluous samples 4/4

- **Named entities**:
  - *<TGT>HowNet<TGT> is a Chinese ontology*
  - *<TGT>Stu-1<TGT> have <TGT>road<TGT> with much satisfaction your remarks*
  - *Count Szeehvyni were on board the <TGT>Seri<TGT> rervas*
  - *Charles <TGT>Wye<TGT> Williams, Esq.*
  - *Private Contract under a <TGT>Fiat<TGT> in <TGT>Bank-pose,<TGT>*
  - *in <TGT>rus-sia<TGT> or morocco letter*

Model comparison

Input	Target	(3,3) LSTM	Isol. Transf.
dictiwanf	dictionary.	dictiwanf	dictionary
li'om	from	lrom	from
ooUeotod	collected	collected	collected
Hkewise	likewise	likewise	likewise
twoscvcraU	two severall	two seveall	two several
aiglit .	sights.	sights.	aights.
xcix.	XCIX.	CCX.	XCIX.
Inll"illcr	IntFilter	Iulriller	InFiller
deUcato	delicate	deUcato	delicate
oonaecntivu	consecutive	conseentive	consecutive
Rela∼d	Related	Relaed	Related

Table: Comparison of the predictions from the two best-performing error correction models

Discrepancy w/ Nguyen et al.

- [7] achieves better results with a similar approach (BERT + LSTM)
- Differences:
  - **Meta input features** - origin of sample
    $\rightarrow$ not applicable for generic model
  - Flexible **target entity positioning**:

    twenty#in#number#andjust#then
    in#number#andjust#then#published

    $\rightarrow$ increases size of training data
  - Trained and **optimized** on ICDAR datasets *exclusively*
    - Reduced impact of *domain Specificity*?
    - ED-based **filter** to suppress corrections with ED > 3
  - Recognition of **word boundary errors** is left up to *correction* model

Table of Contents

Sentence tokenization

- How to **split** large sequences into workable chunks?
- In this paper → **sentence tokenization** with **SpaCy**
  …turned out to be a bad idea
  - Good case: *W. Daelemans, J. Zavrel, P. Berck, and S. Gillis.*
  - Bad case: *No ORDERS|| will be|| admitted.|| To-morrow ... .||(By Particular Desire). -The Brigand. And Wif||e! What Wife ? Monday, .. ▪ ▪ 'RICHES. Luke, ...Mr, Ksan. Tuesday, Paul|| Pry.*
- Common approach from related work: **flat max. length**
  → Split target token in middle? Word boundary errors?