Text Normalization

Dimitris Loupas

August 2025

1 Overall approach and logic

The approach of the text normalization task will be to split it into different phases. The goal is to start with rule-based approaches and progressively increase the complexity by incorporating more complex techniques (AI models). For this reason, the task consist of the following Phases:

Phase 0: Data exploration.

Phase 1: Rule-based normalization. (done)

Phase 2: Named Entity Recognition (NER). (done)

Phase 3: Fine-tuned Large Language Model (LLM). (done)

Phase 4: Agent-based. (no time)

The metrics that will be used for evaluating the predictions are Exact Match Accuracy and Token-Level F1 Score.

Note: For this task Phase 0 and Phase 1 are completed, for the rest we discuss how to tackle them.

2 Evaluation Metrics

We consider two simple metrics to compare predictions with the label `CLEAN_TEXT`.

Exact Match Accuracy. For N rows with label strings y_i and predictions $\hat{y}_i$, the exact match accuracy is the fraction of rows where the strings are exactly equal:

$$\mathrm{EM} \;=\; \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{\hat{y}_i = y_i\}.$$

No partial credit is given; a row counts only if the full strings match.

Token-level F1. For each row, split both strings on whitespace into tokens (no extra normalization), and treat them as multisets. Let

$$\text{overlap} \;=\; \sum_{w} \min\left(\text{count}_y(w),\, \text{count}_{\hat{y}}(w)\right),$$

$$\text{precision} \;=\; \frac{\text{overlap}}{\text{\#pred tokens}}, \quad \text{recall} \;=\; \frac{\text{overlap}}{\text{\#label tokens}},$$

$$\text{F1}_i \;=\; \frac{2\,\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (\text{defined as } 0 \text{ if the denominator is } 0).$$

The final score is the mean of F1_i over all rows.

3 Phase 0 – Data exploration

3.1 Dataset Overview

The dataset `normalization_assesment_dataset_10k.csv` contains **10,000 rows** and 2 columns:

- `raw_comp_writers_text` – original raw text
- `CLEAN_TEXT` – normalized text

3.2 Basic Statistics

- Dataset size: $10{,}000$ samples
- Rows already normalized: $6{,}323$ (63.2 %)

- Rows that need processing: $3{,}677$ ($36.8\,\%$)
- Missing values per column:
  - `raw_comp_writers_text`: 1
  - `CLEAN_TEXT`: $1{,}341$
- Number of rows with NaN: $1{,}341$ ($13.4\,\%$)
- Total missing values: $1{,}342$ ($6.7\,\%$)

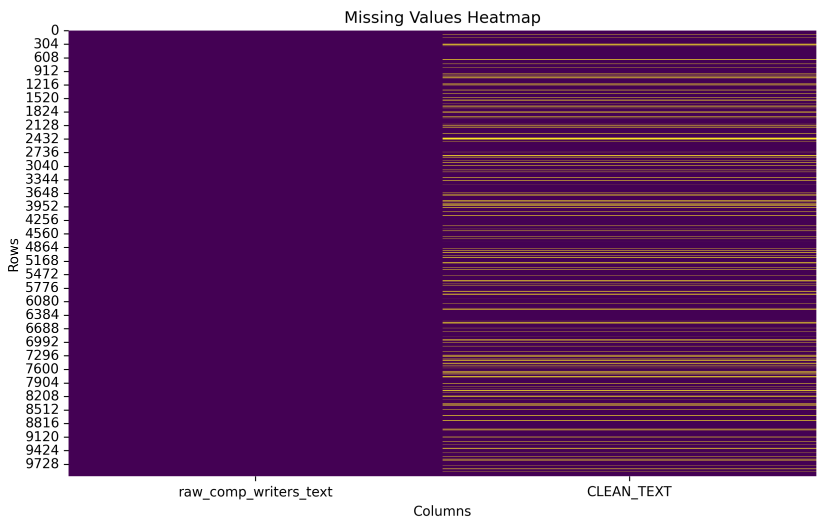

Figure 1: Heatmap of missing values in the dataset.

3.3 Letter Analysis

category	raw_letters	clean_letters	reduction_%
Latin	234,729 (97.95 %)	195,205 (99.96 %)	16.84
Other	4,909 (2.05 %)	77 (0.04 %)	98.43

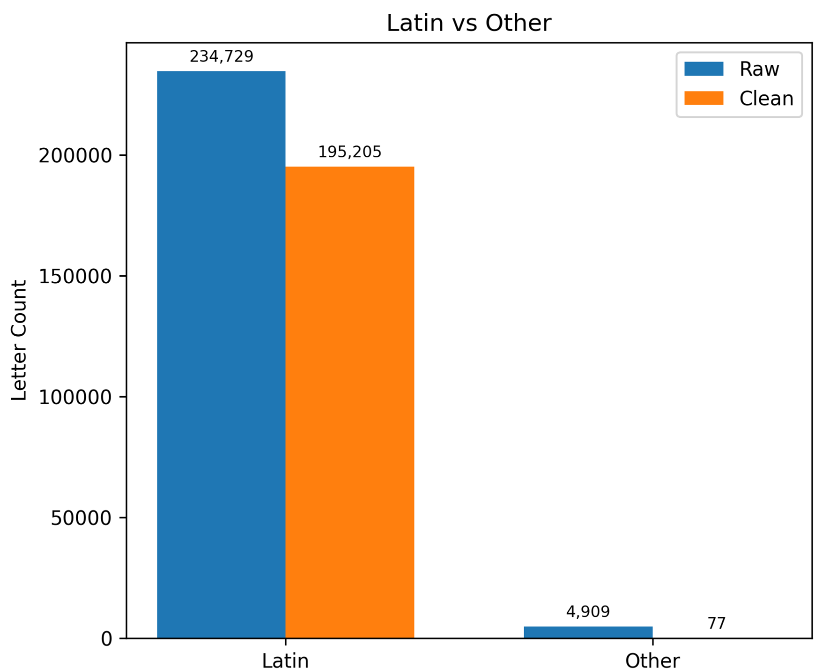

Figure 2: Latin vs Other letter distribution.

3.4 Separator Analysis

1-gram

separator	total_occ_raw	rows_with_raw	total_occ_clean	rows_with_clean
/	7773	3998 (39.98 %)	9060	4152 (41.52 %)
.	986	611 (6.11 %)	741	466 (4.66 %)
&	981	961 (9.61 %)	137	133 (1.33 %)
<	344	342 (3.42 %)	1	1 (0.01 %)
>	344	342 (3.42 %)	2	2 (0.02 %)
-	319	290 (2.90 %)	227	213 (2.13 %)
(	271	213 (2.13 %)	54	45 (0.45 %)
)	263	206 (2.06 %)	54	44 (0.44 %)
'	91	81 (0.81 %)	74	66 (0.66 %)
"	44	17 (0.17 %)	24	11 (0.11 %)
’	33	32 (0.32 %)	27	27 (0.27 %)
“	29	23 (0.23 %)	24	19 (0.19 %)
”	29	23 (0.23 %)	24	19 (0.19 %)
[	25	25 (0.25 %)	2	2 (0.02 %)
]	25	25 (0.25 %)	1	1 (0.01 %)
	20	16 (0.16 %)	14	12 (0.12 %)
#	19	16 (0.16 %)	4	3 (0.03 %)
$	17	15 (0.15 %)	16	15 (0.15 %)
-	16	16 (0.16 %)	14	14 (0.14 %)
:	11	10 (0.10 %)	8	7 (0.07 %)

2-gram

separator	total_occ_raw	rows_with_raw	total_occ_clean	rows_with_clean
>/	343	342 (3.42 %)	1	1 (0.01 %)
)/	62	49 (0.49 %)	30	22 (0.22 %)
./	39	38 (0.38 %)	34	33 (0.33 %)
.)	5	5 (0.05 %)	1	1 (0.01 %)
-/	4	4 (0.04 %)	3	3 (0.03 %)
'/	3	3 (0.03 %)	4	4 (0.04 %)
&/	2	2 (0.02 %)	93	91 (0.91 %)
!/	2	2 (0.02 %)	1	1 (0.01 %)
&#	2	2 (0.02 %)	1	1 (0.01 %)
/\	2	2 (0.02 %)	1	1 (0.01 %)
″/	1	1 (0.01 %)	1	1 (0.01 %)

.:	1	1 (0.01 %)	1	1 (0.01 %)
\/	1	1 (0.01 %)	1	1 (0.01 %)
"/	1	1 (0.01 %)	1	1 (0.01 %)
.”	1	1 (0.01 %)	1	1 (0.01 %)

3-gram

separator	total_occ_raw	rows_with_raw	total_occ_clean	rows_with_clean
.)/	3	3 (0.03 %)	1	1 (0.01 %)
/\/	1	1 (0.01 %)	1	1 (0.01 %)

3.5 Frequent words

1-gram

RAW 1-grams

ngram	count	pct %
unknown	384	0.93
ca	309	0.75
music	227	0.55
john	190	0.46
david	189	0.46
james	183	0.44
michael	181	0.44
copyright	158	0.38
control	154	0.37
pa	153	0.37
de	136	0.33
the	136	0.33
thomas	130	0.31
daniel	116	0.28
paul	109	0.26
robert	103	0.25
publishing	98	0.24
mark	95	0.23
chris	85	0.21
william	84	0.20

CLEAN 1-grams

ngram	count	pct %
david	170	0.51
john	166	0.50
james	166	0.50
michael	161	0.48
thomas	126	0.38
de	120	0.36
the	109	0.33
daniel	109	0.33
paul	101	0.30
robert	96	0.29
mark	86	0.26
pa	83	0.25
chris	77	0.23
william	75	0.22
martin	73	0.22
lee	73	0.22
brown	72	0.21
andrew	71	0.21
mike	70	0.21
christopher	69	0.21

2-gram

RAW 2-grams

ngram	count	pct %
copyright control	150	0.36
music publishing	39	0.09
wolfgang amadeus	20	0.05
amadeus mozart	20	0.05
de la	18	0.04
unknown writer	14	0.03
giuseppe verdi	14	0.03
universal music	14	0.03
johann sebastian	13	0.03
gmbh co	13	0.03
sebastian bach	12	0.03
juice wrld	12	0.03
music gmbh	12	0.03
co kg	12	0.03
warner chappell	11	0.03
sony atv	11	0.03
ludwig van	10	0.02
van beethoven	10	0.02
the alchemist	10	0.02
paul mccartney	9	0.02

CLEAN 2-grams

ngram	count	pct %
wolfgang amadeus	20	0.06
amadeus mozart	20	0.06
de la	15	0.04
giuseppe verdi	14	0.04
johann sebastian	12	0.04
sebastian bach	11	0.03
juice wrld	11	0.03
ludwig van	9	0.03
van beethoven	9	0.03
paul mccartney	9	0.03
the alchemist	9	0.03
billie ray	9	0.03
ray fingers	9	0.03
bruce fingers	9	0.03
thomas bergersen	9	0.03
anand bakshi	9	0.03
tee grizzley	8	0.02
john lennon	8	0.02
lorenzo da	8	0.02
da ponte	8	0.02

3-gram

RAW 3-grams

ngram	count	pct %
wolfgang amadeus mozart	20	0.05
johann sebastian bach	12	0.03
gmbh co kg	11	0.03
ludwig van beethoven	10	0.02
music gmbh co	10	0.02
billie ray fingers	9	0.02
universal music publishing	9	0.02
lorenzo da ponte	8	0.02
tee grizzley skilla	7	0.02
grizzley skilla baby	7	0.02
atv music publishing	7	0.02
jesús maría corman	6	0.01
andrew lloyd webber	6	0.01
warner chappell music	6	0.01
sonoton music gmbh	6	0.01
rahul dev burman	6	0.01
music publishing ltd	5	0.01
pyotr ilyich tchaikovsky	5	0.01
sony atv music	5	0.01
tips industries ltd	5	0.01

CLEAN 3-grams

ngram	count	pct %
wolfgang amadeus mozart	20	0.06
johann sebastian bach	11	0.03
ludwig van beethoven	9	0.03
billie ray fingers	9	0.03
lorenzo da ponte	8	0.02
tee grizzley skilla	7	0.02
grizzley skilla baby	7	0.02
andrew lloyd webber	5	0.01
pyotr ilyich tchaikovsky	5	0.01
rahul dev burman	5	0.01
youngboy never broke	5	0.01
never broke again	5	0.01
john lennon paul	4	0.01
lennon paul mccartney	4	0.01
ray fingers bruce	4	0.01
fingers bruce fingers	4	0.01
big sad 1900	4	0.01
antoine katoto luhembe	4	0.01
jesús maría corman	4	0.01
lankinen mikko juhani	4	0.01

3.6 Frequent words not in CLEAN

1-gram

ngram	count	pct %
bmi	32	1.02
ascap	20	0.64
universal	18	0.57
gmbh	18	0.57
sony	14	0.44
llc	13	0.41
kg	12	0.38
prs	8	0.25
александр	8	0.25
company	7	0.22
андрей	7	0.22
□□□	7	0.22
□□□	7	0.22
limited	6	0.19
audio	6	0.19
industries	6	0.19
corp	6	0.19
sonoton	6	0.19
dp	6	0.19
олег	5	0.16

2-gram

ngram	count	pct %
universal music	14	0.15
gmbh co	13	0.14
music gmbh	12	0.12
co kg	12	0.12
sony atv	11	0.11
writer unknown	8	0.08
unknown brown	8	0.08
music bmi	8	0.08
chappell music	6	0.06

music company	6	0.06
unknown jackson	6	0.06
sonoton music	6	0.06
composer author	5	0.05
tips industries	5	0.05
industries ltd	5	0.05
music ascap	5	0.05
thomas ca	5	0.05
obo gema	4	0.04
unknown rodriguez	4	0.04
music group	4	0.04

3-gram

ngram	count	pct %
gmbh co kg	11	0.09
music gmbh co	10	0.08
universal music publishing	9	0.07
warner chappell music	6	0.05
sonoton music gmbh	6	0.05
music publishing ltd	5	0.04
sony atv music	5	0.04
tips industries ltd	5	0.04
publisher unknown writer	4	0.03
unknown writer unknown	4	0.03
unknown antoine katoto	4	0.03
co kg figurata	4	0.03
kg figurata music	4	0.03
figurata music gmbh	4	0.03
billionaire minds group	4	0.03
acuff rose music	4	0.03
copyright control unknown	3	0.02
publishing bmi adminstered	3	0.02
bmi adminstered by	3	0.02
unknown monti daniel	3	0.02

From those we can obtain a list of stopwords that will help us normalize the text. Some stopwords are ('acuff', 'acuff rose', 'acuff rose music', 'aepi', 'america', 'america obo', 'anh việt thu', 'ar haavisto janne',bmi adminstered', 'bmi adminstered by', 'buddy eden', 'bv', 'ca assaf youhanna', 'ca calcagni filippo', 'ca dixon lance', 'ca granberg marcus'). Moreover in the following analysis I have dropped the rows with missing valus. This raises the precentage of the normalized already rows due to the smaller dataset. The dataset will be:

- Dataset size: $8{,}659$ samples
- Rows already normalized: $6{,}323$ ($73\,\%$)
- Rows that need processing: $3{,}677$ ($27\,\%$)
- No missing values

4 Phase 1 – Rule-Based Normalization

In this task I implemented a rule-based normalization pipeline by considering the results of the previous analysis. I start by stripping out non-Latin noise so not to encounter random symbols. Then I build a blacklist (stopwords found in the previous section) of publisher/metadata phrases: I take my raw ngrams, sort them longest-first (so "sony atv music publishing" gets removed before a shorter "publishing"), escape any regex characters, and glue them into one big word-bounded pattern so I only hit whole terms, not partials. Next comes normalization: I drop anything inside <...>, shave off a leading slash, and standardize separators by turning & and the word and into /, converting commas to /, and collapsing accidental // into a single /. With the text tidy, I run the stopwords regex to delete any publisher/metadata phrases in a case-insensitive pass. Finally, I reduce multiple slashes to one, then trim spaces and any slashes at the ends so the field stays clean and consistent. This approach is fast, transparent, and reproducible, and it achieved Exact Match **Accuracy = 87.33%** and **Token-level F1 = 92.03%**.

5 Phase 2 – Named Entity Recognition

Next, I tried adding a spaCy NER pass on top of the rule-based output. I ran en_core_web_sm over the raw strings, pulled only PERSON spans, and lightly cleaned them (collapse slashes/whitespace and strip dangling connectors like &, and, /, „ +). For each row, I split the Phase-1 result on / and cleaned those pieces too. Then I aligned NER names to the closest Phase-1 variant using a similarity check so near-duplicates share the same canonical name. After that I built a true union: NER names first, then Phase-1 names (case-insensitive). To avoid partial duplicates (e.g., "John" vs "John A. Smith"), I sorted candidates by token-set size and length, kept the longer/more complete forms, and dropped any candidate whose tokens are a subset of one I'd already kept. I then ordered the remainders by their first appearance in the original raw string (so the output follows the natural reading order), joined with /, collapsed any repeated slashes, and fell back to Phase-1 if the list ended up empty. This combo aimed to increase recall without losing the canonical formatting from Phase-1, but in practice the extra NER names introduced enough noise that exact matches fell: Exact **Match Accuracy = 83.02%**, **Token-level F1 = 90.73%**, both slightly worse than the rule-only baseline (87.33% / 92.03%).

6 Phase 3: LoRA Fine-Tuning of a Seq2Seq Model (T5)

Goal. Train a text-to-text model that maps `noisy_text` to `normalized_text`. We use T5 with Low-Rank Adaptation (LoRA) so we can fine-tune fast and with less GPU memory. We start with `t5-small` and, if resources allow, move to `t5-base` or `t5-large` (larger models are expected to help).

6.1 Data and Task Format

- Input CSV has two columns: `noisy_text` and `normalized_text`.

- We split into train/validation (e.g., 80/20).
- We add a task prefix to each input: ``normalize: '' + noisy_text.

6.2 Tokenization

- Tokenizer: T5Tokenizer.
- Truncation/padding to a fixed max length (e.g., 128).
- Labels are the token IDs of normalized_text.

6.3 Model and LoRA

- Base model: T5ForConditionalGeneration.
- Apply LoRA to attention projections (e.g., q, v); freeze the rest.
- Typical LoRA hyperparameters: rank $r \in \{4, 8, 16\}$, lora_alpha $\in \{16, 32\}$, lora_dropout $\in \{0.0, 0.05\}$.

6.4 Training

- Trainer: HuggingFace Trainer with TrainingArguments.
- Example settings: num_train_epochs = 3, small batch_size (increase if VRAM allows), evaluation_strategy = ``epoch'', save_strategy = ``epoch'', load_best_model_at_end = True.
- Optimization focuses on LoRA params only (cheap to train).

6.5 Validation and Metrics

- We compute Exact Match Accuracy (EM) and token-level F1 on the validation set (same as earlier phases) to compare models fairly.

- We keep the best checkpoint by validation loss or EM.

6.6 Inference

- At test time, form the input as ``normalize: '' + text.
- Use `model.generate` and then decode the output.
- Simple decoding works (greedy); beam search can be tried later.

6.7 Notes

- If you have more GPU memory, try `t5-base` or `t5-large`; expect better accuracy.
- Keep the same metrics (EM, token F1) so results are comparable to Phase 1/2.

7 Phase 4 - Agent Based

In the final phase the system uses a clear, agent-style pipeline with four simple roles. Agent 1 (Preprocessor) standardizes the raw text (basic cleanup only). Agent 2 (Rules Agent) applies the Phase-1 rule-based normalizer as code and produces a candidate list of names. Agent 4 (Evaluator/Controller) compares this candidate to the label (CLEAN_TEXT) using exact match and token F1; if it is already correct, the process stops for that row. If it is wrong, the Evaluator sends the row to Agent 3 (Model Agent), which runs a transformer and returns a strict JSON list of names. The Evaluator then compares "Rules vs Model" against the label and chooses one output only (no union); if one is an exact match, it wins, otherwise the one with higher token overlap with the label is selected (ties go to Rules). The model is gated: it runs only when the rules are wrong. The loop is bounded to at most two rounds and stops early as soon as the score improves; if nothing helps, the system falls back to the rule output to avoid getting worse than Phase-1. A lightweight Logger records per-row details (raw text, label, rule output, model output, scores, final choice) for debugging and small ablations. This keeps the flow simple and reproducible, while still letting the model fix the hard rows.

8 Conclusions

This project was only partially implemented. I finished Phase 1 (rule-based) and Phase 2 (NER), and I sketched plans for Phase 3 (LoRA fine-tuning of T5) and Phase 4 (agent-based). The dataset has some issues in the `CLEAN_TEXT` column (it is not always normalized). Cleaning this column should increase the reported accuracy.

The rule-based method gave solid results (**Exact Match = 87.33%**, **Token F1 = 92.03%**). When I combined rules with NER using a union, accuracy dropped (**Exact Match = 85.06%**, **Token F1 = 91.39%**) because NER added extra noisy spans. For future work, I plan to clean the labels, fine-tune a T5 model with LoRA on the full data, and use an evaluator-controlled agent loop that chooses the single best output (no union).