Neural Text Normalization for Luxembourgish Using Real-Life Variation Data

Anne-Marie Lutgen[1], Alistair Plum[1], Christoph Purschke[1], Barbara Plank[2,3],

[1]University of Luxembourg, Esch-sur-Alzette, Luxembourg
[2]MaiNLP, Center for Information and Language Processing, LMU Munich, Germany,
[3]Munich Center for Machine Learning (MCML), Munich, Germany,

Correspondence: anne-marie.lutgen@uni.lu

Abstract

Orthographic variation is very common in Luxembourgish texts due to the absence of a fully-fledged standard variety. Additionally, developing NLP tools for Luxembourgish is a difficult task given the lack of annotated and parallel data, which is exacerbated by ongoing standardization. In this paper, we propose the first sequence-to-sequence normalization models using the ByT5 and mT5 architectures with training data obtained from word-level real-life variation data. We perform a fine-grained, linguistically-motivated evaluation to test byte-based, word-based and pipeline-based models for their strengths and weaknesses in text normalization. We show that our sequence model using real-life variation data is an effective approach for tailor-made normalization in Luxembourgish.

1 Introduction

Automatic text normalization is the task of mapping non-standard spellings to a standard (Han and Baldwin, 2011; van der Goot, 2019). Normalization thus reduces orthographic variation and noise in language data. It can serve as a pre-processing step to facilitate downstream tasks like POS-tagging and NER (e.g. Küçük and Steinberger, 2014; van der Goot and Çetinoğlu, 2021).

In this paper, we address orthographic normalization for Luxembourgish, a Germanic language currently in the process of political development, including orthographic standardization (Gilles, 2019). Spelling norms for Luxembourgish are not a novelty, however, due to a lack of language teaching in school, written Luxembourgish today is characterized by vast amounts of variation, e.g., orthographic, lexical, syntactical or regional. This has led to written Luxembourgish texts adhering to the standard orthography to be rare, even in formal contexts. For this reason, we develop an automatic text normalization model for Luxembourgish to reduce variation in written data as a pre-processing step for NLP tasks.

Luxembourgish is an under-researched language, and as such there is a lack of annotated parallel data for training and fine-tuning normalization models. To tackle this problem, our proposed solution uses word-level real-life variation data to create training data sequences and fine-tune multilingual sequence-to-sequence models. In this paper, we use ByT5 (Xue et al., 2022) and mT5 (Xue et al., 2021) models and in addition, we benchmark the generative models GPT-4o and Llama and a word-based Luxembourgish correction pipeline, *spellux*.[1]

We evaluate model performance using both quantitative metrics and a tailored qualitative evaluation of linguistic contexts. Then, we compare byte-based, word-based and pipeline-based models to identify linguistic contexts in which models perform particularly well or struggle.

Our main contributions are therefore twofold:

(1) The first generative normalization model for Luxembourgish, trained on real-life variation data obtained from an online spellchecker.

(2) A linguistically-informed qualitative test set tailored for Luxembourgish orthography, besides a comprehensive quantitative evaluation.

2 Related Work

Luxembourgish, a relatively small language, is less represented in NLP compared to its linguistic neighbors, French and German. Research in NLP for Luxembourgish has only recently gained momentum, with a few earlier works: Adda-Decker et al. (2008) introduced various resources for NLP tasks in Luxembourgish; Snoeren et al. (2010) analyzed typical writing patterns (contextual n-deletion) in transcribed speech; and Lavergne et al. (2014)

[1]https://github.com/questoph/spellux

Proceedings of the Twelfth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 115–127
January 19, 2025.

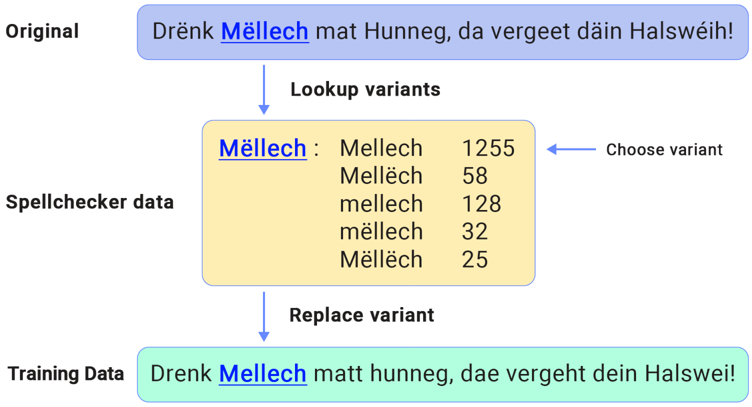

Figure 1: Illustration of the creation of training data with the Luxembourgish Online Dictionary (LOD) sentence 'Drink milk with honey, then your throat will no longer hurt.' and the variational statistical data for 'milk'. The algorithm processes every word sequentially, this illustrates only the replacement process for the word 'milk'.

presented a manually annotated corpus of mixed-language sentences to test a word-based language identification system. Additionally, the first tree-bank for Luxembourgish *Luxbank* was recently released (Plum et al., 2024a).

Other developments include Sirajzade et al. (2020) and Gierschek (2022), who tested various approaches for performing sentiment analysis on Luxembourgish, including training BERT-based models. Philippy et al. (2024) proposed a new approach to zero-shot classification using a task-specific dictionary for topic classification. For spoken data, Gilles et al. (2023a) and Gilles et al. (2023b) developed LUX-ASR, an efficient Automatic Speech Recognition system for Luxembourgish. Additionally, Ranasinghe et al. (2023) fine-tuned language models for automatic comment moderation. Some language models have been trained using transfer learning from German, such as LuxGPT (Bernardy, 2022). Other models developed for Luxembourgish are LuxemBERT (Lothritz et al., 2022) and ENRICH4ALL (Anastasiou, 2022).

Various T5 (Raffel et al., 2020) and ByT5 (Xue et al., 2022) architectures have been developed for lexical normalization. Samuel and Straka (2021) pre-trained a ByT5 model for 12 languages with synthetic data as part of the shared task Multi-LexNorm (van der Goot et al., 2021). Similarly, Rothe et al. (2021) fine-tuned a mT5 model using synthetic parallel data for German, English, Czech and Russian. Kuparinen et al. (2023) evaluated different sequence-to-sequence models including ByT5 for dialect-to-standard normalization in Norwegian, Swiss German, Slovene and Finnish.

Lusetti et al. (2018) developed an encoder-decoder architecture for text normalization in Swiss German by using sequence-to-sequence models but not T5 architectures. Similarly, Bollmann (2018) worked on historical text normalization, comparing encoder-decoder architectures to statistical machine translations.

For pipeline-based normalization, van der Goot (2019) developed MoNoise, which has long been regarded as the state-of-the-art text normalization tool. This pipeline operates at a word level using a spelling correction module and word embeddings for various languages. Furthermore, van der Goot (2019) introduced the error reduction rate (ERR) as an evaluation metric for normalizers. MoNoise has also been used on the task of nested named entities in Danish (Plank et al., 2020) and for code-switching data (van der Goot and Çetinoğlu, 2021). For Luxembourgish, Purschke (2020) published *spellux* a pipeline for automatic orthographic correction of text data, the first text normalization tool for Luxembourgish.

3 Methodology

This section describes the methodology for fine-tuning and evaluating our normalization model for Luxembourgish. This includes the creation of training data, the experimental setup for the model training and benchmarking, and the model evaluation process.

3.1 Creating Training Data

The lack of annotated and parallel datasets in Luxembourgish is a challenge for developing tailored NLP solutions. This not only applies to text normalization but also, for example, to NER and machine translation tasks. Creating parallel datasets manually or via crowdsourcing is not a viable option for Luxembourgish, as the majority of the population has no formal training in orthography due to the lack of extensive grammar teaching in school contexts. Luxembourgish has only recently been integrated more into the school curriculum to foster the societal anchoring of the spelling rules. As a consequence, corpus data for Luxembourgish written in the standard orthography are scarce. The solution applied in this paper is to create training data based on real-life variation data obtained from an online spellchecking tool.

Creating training data from synthetic data for normalization has been used on multiple occasions, as in Samuel and Straka (2021) and Rothe et al. (2021). However, using only synthetic data may be problematic as it does not accurately represent real-life language use. For this paper, we use data provided by the spelling correction website *Spellchecker.lu*[2] as a basis for the creation of training data. Users on the website can manually correct written Luxembourgish text based on context-sensitive suggestions offered by the system. Pairs of entered and corrected words are then logged and statistically aggregated. As a consequence, this dataset offers a unique real-life dictionary of spelling variants per lemma, including their frequency of use.

Using real-life variation to create training data ensures that each variant in the data has actually been used by people and is not just a random character replacement. Additionally, the frequency of use of these variants can be represented in the data realistically. Since the baseline for the replacements mirrors a realistic distribution of spelling variants based on actual texts written by people, this approach is considered superior to using synthetic data. The training data, hence, captures the actual patterns of variation in Luxembourgish and is not a randomly assembled approximation of non-standard texts.

As the website is widely known and used in the country,[3] Spellchecker.lu provides us with an extensive overview of the orthographic variation space in Luxembourgish. The variant dictionary includes 138,802 different lemmas with numerous variants per lemma. Figure 1 illustrates an example with the word *Mëllech* ('milk') and its most frequent variations in the Spellchecker data.

We use transcriptions of discussions in the Chamber of Deputies in Luxembourg as a source of orthographically correct Luxembourgish for the training data.[4] These transcriptions are from 2002-2012 and 2019-2020 and are produced by trained writers, ensuring the correctness of the texts.

Combining the Spellchecker.lu variant dictionary and the transcriptions then allows for the creation of parallel training data using correct Luxembourgish texts and real-life variation patterns. We apply an algorithm that processes every word in the original sentence sequentially and looks up the variants in the Spellchecker.lu data. If the lemma is part of the variant dictionary, it is replaced by a variant based on its frequency of use. This process results in around 833,000 parallel standard/non-standard sentence pairs with a mean token difference of 19 tokens being changed per sentence pairs that can be used as training data. Figure 1 illustrates this process with an example sentence taken from the Luxembourgish Online Dictionary (LOD).[5]

3.2 Experimental Setup

We fine-tune two multilingual sequence-to-sequence models, ByT5 and mT5. For benchmarking the task, we prompt Llama[6] and GPT[7], as well as using the word-based normalization tool *spellux*. In this way, we test various approaches for Luxembourgish text normalization, i.e., a byte-based sequence-to-sequence and word-based sequence-to-sequence method, as well as generative models and a word-based pipeline. Due to a lack of training data we did not opt for pre-training a sequence-to-sequence model ourselves for this task. However, recently Plum et al. (2024b) pre-trained a T5-based model with multilingual data to improve performance for Luxembourgish.

ByT5 is a multilingual byte-based sequence-to-sequence model which encodes the input sequence to UTF-8-encoded bytes and produces an output sequence of UTF-8-encoded bytes (Xue et al., 2022). The robustness to noise and variation of byte-based and character-based models (Xue et al.,

[2] https://spellchecker.lu

[3] The popularity of the website can be explained by the recent increase in the use of written Luxembourgish in the population, without having formal orthography training. The website provides a helpful tool for writing Luxembourgish correctly, e.g., in formal contexts.

[4] https://www.chd.lu/de/chamberblietchen

[5] https://lod.lu

[6] Llama-3.1-8B-Instruct

[7] gpt-4o-2024-08-06

2022) makes them ideal models to fine-tune for normalization in Luxembourgish. mT5 is a multilingual transformer encoder-decoder model trained on 101 languages (Xue et al., 2021). While ByT5 is the focus of our experiment, the word-based model mT5 is used as a comparison to the byte-based approach for a sequence-to-sequence task.

The fine-tuning setup stays the same for the ByT5 and mT5 models. Our experiments focus on testing the experimental method (real-life training data and comprehensive performance testing) rather than producing an optimized model, although we did perform some hyperparameter tuning, where we were not constrained by hardware limitations. Using the ByT5 base model with 582M parameters, the best performing model has a batch size of 16, a learning rate of 1e-4, and a sequence length of 256 trained on 3 epochs. We also fine-tune the ByT5 large model with 1.23B parameters to test the influence of parameter size on task performance. We restrict the hyperparameter setup for fine-tuning to a sequence length of 128 and an epoch number of 1. The mT5 model is fine-tuned using the base variant with 582M parameters. Additionally, we fine-tune one mT5 model with the same hyperparameters as the ByT5 model, a sequence length of 128, 1 epoch and batch size 2.

Benchmarking is done by prompting GPT-4o and Llama 3.1. The setup is the same for both models, using the following prompt: "You are a Luxembourgish teacher. Your task is to correct these sentences on a word level based on the correct Luxembourgish orthography. Please only write the corrected sentence and no explanation". For this task, we use the same evaluation sentences as for the other models. The main focus of the setup lies on models that are developed specifically for Luxembourgish, nonetheless we include GPT-4o and Llama for completeness reasons as the results are not reproducible and there is a lack of knowledge concerning the training data (see Section 5).

Our main comparison of the models is with the *spellux* text correction pipeline, which is developed specifically for Luxembourgish. The tool implements a combination of correction algorithms for candidate evaluation: a word-based embedding model trained on the entire archive from RTL.lu (journalistic texts and user comments), an adapted version of the spelling correction tool written by Peter Norvig[8], and an ngram-based tf-idf similarity matrix based on the RTL corpus. *spellux* also includes an adapted version of the variant dictionary created from the Spellchecker data. For benchmarking, we use the default settings of the pipeline.

3.3 Evaluation

We perform a comprehensive evaluation of the models based on both quantitative metrics and a qualitative analysis, where we compare the output of different models to gain more insight into how well the different models solved the task. First, we perform a quantitative evaluation using a wide array of evaluation metrics, then, we develop a set of qualitative tests tailored to the normalization task, inspired by CheckList (Ribeiro et al., 2020). This allows for a linguistically informed and systematic analysis of the output and performance of the used models.

Quantitative For the quantitative evaluation, we create a corpus consisting of random user comments from the RTL media platform, as they contain a high amount of variation (Purschke, 2020), and correct them manually. This results in an evaluation corpus consisting of 459 sentences from the comments, equalling 7,146 tokens.

The evaluation metrics used for the fine-tuned models and benchmarking include accuracy, recall, precision, F1-score, and error reduction rate (ERR) at the word-level, and character error rate (CER) at the character level. To calculate the word-level metrics, we align the original sentences, the predicted output sentences and the orthographically correct sentences at the word-level. The alignment is done by repurposing the *Needleman-Wunsch* algorithm (Needleman and Wunsch, 1970) with Kamil Slowikowski's code for 3D alignment and string alignment[9], using the Levenshtein distance for fuzzy string matching. Although other distance metrics are available, we did not carry out any experiments with these as this was not within the scope of our research. We therefore opted for the Levenshtein distance, since it is well established.

The most important metric for the normalization task is the ERR, introduced by van der Goot (2019) as an evaluation metric for normalizers, and used as the main evaluation metric in van der Goot et al. (2021). It captures the accuracy normalized over the number of words to be corrected (van der Goot, 2019). The ERR normally has a value between 0

[8] `https://norvig.com/spell-correct.html`

[9] `https://gist.github.com/slowkow/06c6dba9180d013dfd82bec217d22eb5`

Category	Test Sentence
Quantity rule	
writing of long vowels depending on stressed vowels & consonants (Gilles, 2015)	Wou ass d'Bischt fir ze kieren? *Correction:* Biischt *Where is the broom to sweep (with)?*
Short Vowels	
stressed short vowels and consonants (Gilles, 2023)	D'Haus ass op mech geschriwen. *Correction:* geschriwwen *The house is written under my name.*

Table 1: Selection of test units for Luxembourgish. Full set of rules with examples provided in the Appendix.

and 1. Zero represents the leave-as-is baseline, a negative value indicates that the model performs worse than the baseline, and a positive value means that the model normalizes more words correctly. The comparability across multiple corpora is the main advantage of using this metric, as the ERR is a normalized value (van der Goot, 2019).

Besides word-level metrics, the character-level metric CER is included so that the evaluation becomes more granular. This means being able to not only distinguish between words that are either simply correct or incorrect, but also by how many characters words have changed (Kuparinen et al., 2023). While this is by no means an indicator for degrees of correctness, the metric does allow for gauging how far away a predicted sentence is from its correct form.[10]

Qualitative For the qualitative evaluation, we use a setup similar to CheckList (Ribeiro et al., 2020), a methodology to systematically test NLP models, to evaluate the performance of the normalizer. Specifically, we use the Minimum Functionality test to probe the model as to the handling of Luxembourgish orthographic rules and to gain more linguistic insights into the strengths and weaknesses of the different models. These tests include two different setups and implement 21 orthographic rules. These rules are implemented based on the official Luxembourgish orthography.[11] The first setup tests the traditional application of a normalizer by correcting an incorrect target word, therefore checking corrections systematically against the backdrop of orthographic rules. The target word is corrupted systematically by applying the orthographic rule in reverse.

The second setup tests false positives by giving a correct input and examining the number of false corrections proposed by the model. We include this test because of the known issues with automatic text normalization, which might increase the number of incorrect forms in a given text. This is also captured in the ERR, as a value under 0 indicates more mistakes than before. We include 10 sentences per test setup per category, which results in 420 sentences.[12]

Table 1 shows selected categories, with a short description and a test sentence from the first setup. Appendix A includes the full table with the 21 rules following the same format. The tables also include references to linguistic literature for each respective phenomenon. The first category in Table 1 is the *quantity rule* which describes the use of the long vowels <a, i, o, u, ä, ö, ü>. The test sentence stems from the first setup and the underlined word *Bischt* ('broom') is the target word, that the model should correct into the correct form *Biischt*.

4 Results

This section illustrates the results from the comprehensive quantitative evaluation and the linguistically-informed qualitative tests. The evaluated models include fine-tuned ByT5 and mT5 models, generative models GPT and Llama and the pipeline-based *spellux*.

4.1 Quantitative

Table 2 shows all the models trained on the normalization task for Luxembourgish, including the benchmarking with GPT, Llama and *spellux*. It is a comparison between a byte-based, word-based, generative-based and pipeline-based normalization method for Luxembourgish. As already established,

[10] The CER is calculated using the implementation available at `https://github.com/nsmartinez/WERpp` following Kuparinen et al. (2023)

[11] D'Lëtzebuerger Orthografie, ZLS 2022.

[12] All sentences are taken from the LOD.

Model	Accuracy	Recall	Precision	F1-Score	ERR	CER
ByT5 base	78.8	54.8	65.9	59.8	**0.26**	11.7
ByT5 large	71.8	51.3	49.6	50.4	-0.01	20.4
mT5	27.6	35.5	5.7	9.7	-5.70	22.2
Llama	63.7	0.0	0.0	0.0	-0.15	10.7
GPT-4o	84.8	66.0	77.5	71.3	**0.46**	7.2
spellux	82.2	46.8	86.3	60.7	**0.39**	7.5

Table 2: Evaluation of models, scores are in percentages except ERR.

the ERR is the most important metric for normalization.

The ByT5 base model is the best performing model using T5 architecture for Luxembourgish, with an ERR of 0.26, an accuracy of 78.79% and a precision of 65.9%, taking into account that this model is pre-trained on multilingual data. In comparison, ByT5 large, for which we did not perform any hyperparameter optimization, only reproduces the leave-as-is baseline with an ERR of -0.01. In contrast, mT5 performs the worst among the T5 architectures. Accuracy and precision are very low, as is the ERR. Additionally, the CER is the lowest for the ByT5 base model, indicating fewer mistakes in a corrected corpus than in other models. Hence, the byte-based model is more suitable for Luxembourgish text normalization than the other tested models.

Recurring issues with both ByT5 models are hallucination, including the repetition of training data and stopping early with long sentences. Kuparinen et al. (2023) encountered similar issues with stopping early. However, an increase in epochs and sequence length when training the ByT5 base model reduces the hallucination rate to 5% and the stopping rate to 2%.

The benchmarked generative models perform very differently from each other. Llama shows an even worse performance than mT5 with an ERR of -0.15. The accuracy of 63.7% is not much lower than the other models, but Llama achieves 0 true positives and therefore a F1-score of 0. In comparison, GPT-4o performs well, with the highest ERR score for this task. An important factor to consider is the rapid progress of GPT and therefore the issue of reproducibility with these generative models. The benchmarking results using the 3 month older gpt-4o-2024-05-13 are much lower than the current results with an ERR of 0.12. This demonstrates how quickly GPT has improved, albeit with a lack of transparency.

In contrast, the pipeline-based model *spellux* has a good performance overall. In particular, the high ERR rate of 0.39 indicates a high correction rate. Only recall is lower than for the ByT5 base model.

4.2 Qualitative

In a second step, we evaluate ByT5 base, mT5 and the *spellux* pipeline qualitatively, focusing on models that are specifically trained for Luxembourgish, to compare the linguistic performance of each approach: byte-based, word-based and pipeline-based. Table 3 shows the results of this evaluation, with the scores indicating the success rate for the first (*correct* columns) and second (*preserve* columns) test setup. As described in Section 3.3, the first setup tests the correction of target words and the second setup the handling of false positives.

Although ByT5 and *spellux* have the same score in 7 categories, ByT5 performs better in 9 categories. In comparison, *spellux* only performs better than ByT5 in 5 categories. The starkest differences in performance are present in the category <s> and <g>. ByT5 has a success rate of 80% in comparison to the 40% of *spellux* in the <s> category. This category describes the orthographic rule for the unvoiced and voiced <s>, a phenomenon also influenced by the orthography of a related German word. Instead of correcting the incorrect form, *spellux* keeps the input form, creating a false negative, or changing the word into a different word with a different meaning. For instance, in the test sentence with the target word *iesen*, where the correct form would be *iessen* (‘to eat’), it corrects the word to *eisen* (‘ours’). On the other hand, *spellux* has a higher success rate in the category <g> with 80% compared to ByT5 with 30%. This category describes the difference between the realization of <g> as a plosive and as a fricative (Gilles and Trouvain, 2013). When the grapheme is realized as a fricative, the <g> is never doubled, as opposed to when the <g> is realized as a plosive. When looking into the output of ByT5, it can be seen that ByT5 keeps the target word the same instead of

Category	ByT5 base		mT5		spellux	
	correct	preserve	correct	preserve	correct	preserve
Quantity Rule	80	80	20	100	80	100
Short Vowels	70	100	20	100	50	100
Short Open Vowel [æ]	50	100	10	100	50	100
Short Closed Vowel [e]	70	90	20	100	40	100
Neutral Short Vowel [ə]	90	100	**40**	100	70	100
Long Vowel [eː]	40	100	0	100	30	100
Diphthongs	60	100	20	100	50	100
r-Rule	70	100	0	100	60	100
Final Devoicing	60	100	30	90	40	100
Consonants <f, v, w>	10	90	10	100	30	100
Consonant <g>	**30**	100	10	100	**80**	100
Consonants <g, ch>	50	90	0	100	50	100
Consonant <h>	**50**	**70**	20	100	50	100
Consonants <j, sch>	20	100	0	100	20	100
Consonants <k, x>	50	100	10	100	40	100
Consonant <s>	**80**	90	10	100	**40**	100
Consonant <z>	30	100	0	100	40	100
n-Rule	40	90	20	100	40	90
French Loanwords	50	90	20	100	60	90
Silent <e>	20	90	0	100	30	100
Plural French Loanwords	**10**	100	0	100	**10**	100

Table 3: Success rate of Performance Tests, all scores are in percentages. The **correct** columns refer to sentences, where a correction is necessary, the **preserve** columns to sentences that should not be corrected. Results in bold are discussed in Section 4.2.

correcting it, creating a false negative.

Interestingly, both the ByT5 and *spellux* show the same low success rate of 10% with the plural of French loanwords. French and German are both contact languages to Luxembourgish, which allowed for a rich borrowing history from both languages (Conrad, 2023). This resulted in the orthographic inclusion of those words, particularly for French loanwords. Morphologically, the plural forms in Luxembourgish (<-en, -er>) are applied to French loanwords instead of French plural forms (<-s>). Due to the phonological phenomenon of deleting the <-n> ending before specific consonants, the <-e(n)> is replaced with <-ë> to avoid ambiguity (Gilles, 2015). This rule is limited to French loanwords and both ByT5 and *spellux* have a very low score. However, considering that the training data (the Chamber texts) contain many French loanwords – they are frequent in the political domain – it is somewhat surprising that ByT5 does not perform better in this category and *spellux* might have achieved better results using the advanced correction modes.

While ByT5 and *spellux* perform similarly, mT5 shows low scores in every category. This aligns with our expectations based on the low performance in the quantitative evaluation. The best score for mT5 is 40% in the neutral short vowel [ə] category, a frequently realized sound in Luxembourgish (Gilles and Trouvain, 2013). This is the written equivalent of the schwa, which is <e> for an unstressed syllable or <ë> for a stressed syllable.

Overall, the second test setup (*preserve* columns) indicates near perfect scores for all three models. Only ByT5 has a lower score of 70% for the category <h> (vowel lengthening through <h> insertion) which is not used in written Luxembourgish but common in German. The lower performance in this category might be explained by the pre-training of the ByT5 model on different languages, including German. It is possible that false transfer learning from German to Luxembourgish could cause a lower performance.

5 Discussion

Automatic text normalization is a challenging task whose success depends on a number of factors, including a societally-anchored orthographic norm as the target of the correction task, the availability of large and standard-adherent datasets, suitable technical approaches for implementing the task, and a thorough understanding of the respective strengths and weaknesses of each approach. Given the current situation of written Luxembourgish – with a standard under development and limited amounts of correctly spelled text – in this paper we investigate text normalization approaches and present a comprehensive evaluation suite.

One of the challenges in developing text normalization tools is the use of synthetic data. While these are easy to produce based on existing corpora and orthographic rule sets, they do not represent the variation that would occur in real texts. To overcome these shortcomings, we present a new approach to generating training data with real-life variation data derived from actual texts written and corrected by writers of Luxembourgish. This approach has a clear advantage over synthetic data or prompting LLMs, as it represents the variation space of a language realistically, according to the actual writing practices of its speakers. In particular, the combination of variants and their frequency of use allows the creation of training data that reflect the variation patterns found in real-life texts.

Another problem with automatic text normalization is model evaluation. Given the large amount of variation found in written Luxembourgish, our approach to model evaluation includes a comprehensive set of quantitative and qualitative tests. These allow for a more fine-grained and linguistically informed analysis of the model output, e.g. by comparing success rates for specific orthographic rules. In this way, our evaluation suite increases the transparency of traditional evaluation metrics.

The results of the evaluation experiments show that the tested approaches not only perform differently in terms of quantitative success, e.g. ERR, but also show particular strengths and weaknesses for specific orthographic rules and contextual phenomena. In general, the latest version of GPT (October 2024) outperforms all other approaches, both model-based and pipeline-based. At the same time, the ByT5 model presented in this paper and the *spellux* correction pipeline show individual strengths for certain sets of orthographic phenomena. Nevertheless, we believe that working with a technical solution tailored to Luxembourgish can be advantageous. First, our approach allows us to control all aspects of model training, i.e., training data, model parameters, and task implementation. Second, the use of real-life variation data as a basis for model training brings our approach closer to the actual variation space found in writing practice. Third, since the standard orthography is still under development, we can easily adapt and optimize our approach to future versions of the standard. Fourth, by combining linguistic analysis and hyperparameter optimization, our approach offers great potential for future iterations.

Looking beyond the task of text normalization, our approach can also serve as a linguistic analysis tool for detecting and classifying variation patterns in written Luxembourgish, for example in the context of the research project *Tracing Attitudes And Variation In Online Luxembourgish Text Archives* (TRAVOLTA).[13] Using journalistic texts and user comments from the media platform RTL.lu, we can trace the development of individual as well as group-based writing practices outside the official spelling norm. Since there is hardly any research on the development of the written domain in Luxembourgish, the project can contribute to a better understanding of individual writing practices as well as the structure and dynamics of its variation space in general.

6 Conclusion

In this paper, we present the first generative normalization model for Luxembourgish by creating training data from real-life variation data. More importantly, we develop performance tests for this normalizer to achieve a comprehensive, linguistically-informed evaluation using both quantitative and qualitative metrics. For the creation of training data, we use a variant dictionary with frequency information to create parallel training data with incorrect and correct sentence pairs. This training data is then used to fine-tune a ByT5 model and a mT5 model: the first sequence-to-sequence models fine-tuned for this task. Additionally, benchmarking is performed to compare byte-based (ByT5), word-based (mT5), LLM-based (Llama, GPT) and pipeline-based (*spellux*) approaches. Furthermore, performance tests for Luxembourgish text normalization offer a deeper insight into the strengths and weaknesses of the models, as we compare ByT5, mT5 and *spellux*.

As the performance of ByT5 shows, our approach to the generation of training data is an effective method to train models while preserving a realistic variation space in the data. Furthermore,

[13] https://www.uni.lu/fhse-en/research-projects/travolta/

the ByT5 base model achieves comparable performances to other approaches with an ERR of 0.26. Overall, this paper shows that normalization for Luxembourgish is possible and achieves good results, either with prompting LLMs, using an already established pipeline, or with a ByT5 architecture.

Limitations

Due to the lack of a full-fledged standard in Luxembourgish, there is a very broad variation space with overlapping spelling variants. Therefore, the Spellchecker.lu variants should not be taken to reflect all possible variants in the variation space in Luxembourgish as they only reflect the users of the website.

We have limited computing resources concerning specifically GPU space which results in a limited hyperparameter optimization setup. The GPU nodes available and used for the experiments are Dual CPU with 4 Nvidia accelerators and 768 GB RAM.

Acknowledgments

This research was supported by the Luxembourg National Research Fund (Project code: C22/SC/117225699).

We would also like to thank Emilia Milano and Verena Blaschke for their invaluable input, Jacques Spedener for the illustration, Lou Pepin for the Luxembourgish corrections, and Noah Lee for providing the idea for the Needleman-Wunsch algorithm.

References

Martine Adda-Decker, Thomas Pellegrini, Eric Bilinski, and Gilles Adda. 2008. Developments of "Lëtzebuergesch" Resources for Automatic Speech Processing and Linguistic Studies. In *Proceedings of LREC 08*, Marrakech, Morocco. ELRA.

Dimitra Anastasiou. 2022. ENRICH4ALL: A First Luxembourgish BERT Model for a Multilingual Chatbot. In *Proceedings of ELRA/ISCA Special Interest Group on Under-Resourced Languages*, Marseille, France. ELRA.

Laura Bernardy. 2022. A Luxembourgish GPT-2 Approach Based on Transfer Learning. Master's thesis, University of Trier.

Marcel Bollmann. 2018. *Normalization of historical texts with neural network models*. Doctoral thesis, Ruhr-Universität Bochum, Universitätsbibliothek.

François Conrad. 2017. *Variation durch Sprachkontakt*. Peter Lang Verlag, Berlin, Germany.

François Conrad. 2023. *Deutsch-luxemburgischer Sprachkontakt in Luxemburg*, pages 53–88. Olms Verlag.

Daniela Gierschek. 2022. *Detection of Sentiment in Luxembourgish User Comments*. Ph.D. thesis, University of Luxembourg.

Peter Gilles. 2006. Phonologie der n -Tilgung im Moselfränkischen ('Eifler Regel'). Ein Beitrag zur dialektologischen Prosodieforschung. In *Perspektiven einer linguistischen Luxemburgistik. Studien zur Diachronie und Synchronie*. Winter.

Peter Gilles. 2014. Phonological domains in Luxembourgish and their relevance for the phonological system. In *Syllable and Word Languages*. de Gruyter.

Peter Gilles. 2015. From Status to Corpus: Codification and Implementation of Spelling Norms in Luxembourgish. In *Language Planning and Microlinguistics: From Policy to Interaction and Vice Versa*, pages 128–149. Springer.

Peter Gilles. 2019. *39. Komplexe Überdachung II: Luxemburg. Die Genese Einer Neuen Nationalsprache*, pages 1039–1060. De Gruyter Mouton, Berlin, Boston.

Peter Gilles. 2023. Luxembourgish. In *The Oxford Encyclopedia of Germanic Linguistics*. Oxford University Press.

Peter Gilles, Léopold Edem Ayité Hillah, and Nina Hosseini Kivanani. 2023a. ASRLUX: AUTOMATIC SPEECH RECOGNITION FOR THE LOW-RESOURCE LANGUAGE LUXEMBOURGISH. In *Proceedings of the 20th International Congress of Phonetic Sciences*. Guarant International.

Peter Gilles, Nina Hosseini Kivanani, and Léopold Edem Ayité Hillah. 2023b. LUX-ASR: Building an ASR system for the Luxembourgish language. In *Proceedings - 2022 IEEE Spoken Language Technology Workshop (SLT)*.

Peter Gilles and Jürgen Trouvain. 2013. Luxembourgish. *Journal of the International Phonetic Association*, 43(1):67–74.

Peter Gilles and Jürgen Trouvain. 2015. Closure durations in stops and grammatical encoding: On definite articles in Luxembourgish. In *Proceedings of the 18th International Congress of Phonetic Sciences. Glasgow, UK: the University of Glasgow*.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 368–378, Portland, Oregon, USA. Association for Computational Linguistics.

Dilek Küçük and Ralf Steinberger. 2014. Experiments to improve named entity recognition on Turkish tweets. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*, pages 71–78, Gothenburg, Sweden. Association for Computational Linguistics.

Olli Kuparinen, Aleksandra Miletić, and Yves Scherrer. 2023. Dialect-to-Standard Normalization: A Large-Scale Multilingual Evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13814–13828, Singapore. Association for Computational Linguistics.

Thomas Lavergne, Gilles Adda, Martine Adda-Decker, and Lori Lamel. 2014. Automatic language identity tagging on word and sentence-level in multilingual text sources: a case-study on Luxembourgish. In *Proceedings of LREC 2014*, Reykjavik, Iceland. ELRA.

Cedric Lothritz, Bertrand Lebichot, Kevin Allix, Lisa Veiber, Tegawende Bissyande, Jacques Klein, Andrey Boytsov, Clément Lefebvre, and Anne Goujon. 2022. LuxemBERT: Simple and Practical Data Augmentation in Language Model Pre-Training for Luxembourgish. In *Proceedings of LREC 2022*, Marseille, France. European Language Resources Association.

Massimo Lusetti, Tatyana Ruzsics, Anne Göhring, Tanja Samardžić, and Elisabeth Stark. 2018. Encoder-decoder methods for text normalization. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 18–28, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Saul B. Needleman and Christian D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453.

Fred Philippy, Shohreh Haddadan, and Siwen Guo. 2024. Forget NLI, use a dictionary: Zero-shot topic classification for low-resource languages with application to Luxembourgish. In *Proceedings of the 3rd Annual Meeting of the Special Interest Group on Under-resourced Languages LREC-COLING 2024*, Torino, Italia. ELRA and ICCL.

Barbara Plank, Kristian Nørgaard Jensen, and Rob van der Goot. 2020. DaN+: Danish Nested Named Entities and Lexical Normalization. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6649–6662, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Alistair Plum, Caroline Döhmer, Emilia Milano, Anne-Marie Lutgen, and Christoph Purschke. 2024a. LuxBank: The First Universal Dependency Treebank for Luxembourgish. *Preprint*, arXiv:2411.04813.

Alistair Plum, Tharindu Ranasinghe, and Christoph Purschke. 2024b. Text generation models for luxembourgish with limited data: A balanced multilingual strategy. *Preprint*, arXiv:2412.09415.

Christoph Purschke. 2020. Attitudes Toward Multilingualism in Luxembourg. A Comparative Analysis of Online News Comments and Crowdsourced Questionnaire Data. *Frontiers in AI*, 3.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140).

Tharindu Ranasinghe, Alistair Plum, Christoph Purschke, and Marcos Zampieri. 2023. Publish or hold? Automatic comment moderation in Luxembourgish news articles. In *Proceedings of RANLP 2023*, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A Simple Recipe for Multilingual Grammatical Error Correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.

David Samuel and Milan Straka. 2021. ÚFAL at MultiLexNorm 2021: Improving Multilingual Lexical Normalization by Fine-tuning ByT5. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 483–492, Online. Association for Computational Linguistics.

Joshgun Sirajzade, Daniela Gierschek, and Christoph Schommer. 2020. An Annotation Framework for Luxembourgish Sentiment Analysis. In *Proceedings of SLTU-CCURL 2020*, Marseille. LREC.

Natalie D. Snoeren, Martine Adda-Decker, and Gilles Adda. 2010. The study of writing variants in an under-resourced language: Some evidence from mobile n-deletion in Luxembourgish. In *Proceedings of LREC 2010)*, Valletta, Malta. ELRA.

Rob van der Goot. 2019. MoNoise: A Multi-lingual and Easy-to-use Lexical Normalization Tool. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 201–206, Florence, Italy. Association for Computational Linguistics.

Rob van der Goot and Özlem Çetinoğlu. 2021. Lexical normalization for code-switched data and its effect on POS tagging. In *Proceedings of the 16th Conference of the European Chapter of the Association* *for Computational Linguistics: Main Volume*, pages 2352–2365, Online. Association for Computational Linguistics.

Rob van der Goot, Alan Ramponi, Arkaitz Zubiaga, Barbara Plank, Benjamin Muller, Iñaki San Vicente Roncal, Nikola Ljubešić, Özlem Çetinoğlu, Rahmad Mahendra, Talha Çolakoğlu, Timothy Baldwin, Tommaso Caselli, and Wladimir Sidorenko. 2021. MultiLexNorm: A Shared Task on Multilingual Lexical Normalization. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 493–509, Online. Association for Computational Linguistics.

Rob Matthijs van der Goot. 2019. *Normalization and parsing algorithms for uncertain input*. Ph.D. thesis, University of Groningen.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

A Performance Test Units

Category	Test Sentence
Quantity rule writing of long vowels depending on stressed vowels & consonants (Gilles, 2015)	Wou ass d'Bischt fir ze kieren? *Correction:* Biischt *Where is the broom to sweep (with)?*
Short Vowels stressed short vowels and consonants (Gilles, 2023)	D'Haus ass op mech geschriwen. *Correction:* geschriwwen *The house is written under my name.*
Short Open Vowel [æ] distinction between <e> and <ä> (Gilles, 2015)	D'Mässere si frësch geschlaff! *Correction:* Messere *The knives have been sharpened.*
Short Closed Vowel [e] distinction between <e> and <é> (Gilles, 2015)	Meng Wunnéng ass um drëtte Stack. *Correction:* Wunneng *My flat is on the third floor.*
Neutral Short Vowel [ə] distinction between <e> and <ë> for schwa sound (Gilles, 2014)	Kämm der deng Hoër! *Correction:* Hoer *Comb your hair!*
Long Vowel [eː] distinction between <e> and <ee> (Gilles, 2015)	Mäi beschte Frënd ass Chines. *Correction:* Chinees *My best friend is chinese.*
Diphthongs distinction between the 8 diphthongs (Gilles and Trouvain, 2013)	Firwat hues de dat net gleich gesot? *Correction:* gläich *Why didn't you say that straight away?*
r-Rule distinction between consonant <r> and vocalized (Gilles, 2015)	De Poulet ass nach net ganz durch. *Correction:* duerch *The chicken is not quite done yet.*
Final Devoicing distinction of voiced and unvoiced final consonants (Gilles and Trouvain, 2015)	Eise Projet huet eng zolitt Basis. *Correction:* zolidd *Our project has a solid base.*
Consonants <f, v, w> distinction between <f, v, w> based on German (Gilles, 2015)	Du waars e brawe Jong. *Correction:* brave *You were a good boy.*
Consonant <g> distinction between <g> as a plosive and fricative (Conrad, 2017)	Hues du mech op dëser Foto getagt? *Correction:* getaggt *Did you tag me on this photo?*

Table 4: Performance test units (part 1).

Category	Test Sentence
Consonants <g, ch> distinction between writings of fricatives after vowels (Gilles, 2015)	Ech wunnen an der Buerch. *Correction:* Buerg *I live next to the castle.*
Consonant <h> consonant <h> and non-existent expansion <h>	All eis Méih war ëmsoss! *Correction:* Méi *All our effort was for nothing.*
Consonants <j, sch> writing of fricatives (Conrad, 2017)	Am Zuch hunn e puer Leit Kaart geschpillt. *Correction:* gespillt *A few people were playing cards on the train.*
Consonants <k, x> writing of consonants <k,x>	Dat Kand huet e gudde Karakter. *Correction:* Charakter *This child has a good character.*
Consonant <s> distinction of voiced and unvoiced <s> (Gilles, 2015)	Mir iesen de Mëtteg Nuddelen. *Correction:* iessen *We're having pasta for lunch.*
Consonant <z> distinction between <z> and <tz>	Hie geréit fënnef Keele mat enger Klaz. *Correction:* Klatz *He knocked down 5 pins with one ball.*
n-Rule deletion of final <-n> before specific characters (Gilles, 2006)	De Theo war am Orall op Zak. *Correction:* De *Thea was quick to answer in his oral exam.*
French Loanwords writing of French loanwords (Conrad, 2023)	Hues du deng Valise scho gepaakt? *Correction:* Wallis *Have you packed your case already?*
Silent <e> silent <e> of French loanwords (Gilles, 2014)	Ech ginn ni ouni Necessair op d'Rees. *Correction:* Necessaire *I will never go without my sewing kit on vacation.*
Plural French Loanwords plural of French loanwords <-er, -en, -ë, -éen, -éë> (Conrad, 2023)	Mir kréien am Fréijoer nei Faccen. *Correction:* Facen *We are getting a new facade in spring.*

Table 5: Performance test units (part 2).