A Large-Scale Comparison of Historical Text Normalization Systems

Marcel Bollmann
Department of Computer Science
University of Copenhagen, Denmark
marcel@di.ku.dk

Abstract

There is no consensus on the state-of-the-art approach to historical text normalization. Many techniques have been proposed, including rule-based methods, distance metrics, character-based statistical machine translation, and neural encoder–decoder models, but studies have used different datasets, different evaluation methods, and have come to different conclusions. This paper presents the largest study of historical text normalization done so far. We critically survey the existing literature and report experiments on eight languages, comparing systems spanning all categories of proposed normalization techniques, analysing the effect of training data quantity, and using different evaluation methods. The datasets and scripts are made publicly available.

1 Introduction[1]

Spelling variation is one of the key challenges for NLP on historical texts, affecting the performance of tools such as part-of-speech taggers or parsers and complicating users' search queries on a corpus. *Normalization* is often proposed as a solution; it is commonly defined as the mapping of historical variant spellings to a single, contemporary "normal form" as exemplified in Figure 1.

Automatic normalization of historical texts has a long history, going back to at least Fix (1980). Earlier approaches often rely on hand-crafted algorithms tailored to one specific language, while more recent approaches have focused on supervised machine learning, particularly character-based statistical machine translation (SMT) and its neural equivalent (NMT). However, no clear consensus has emerged about the state of the art for this task, with papers either reporting an advantage for NMT (Hämäläinen et al., 2018), SMT

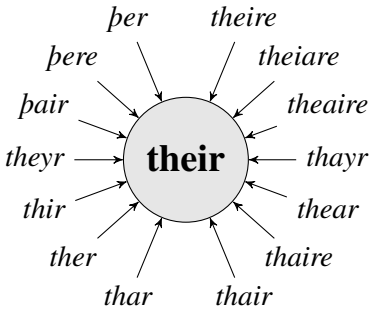


Figure 1: Historical text normalization exemplified: mapping variant spellings from historical English texts to their normalization *'their'*

(Domingo and Casacuberta, 2018), or language-specific algorithms (Schneider et al., 2017). Moreover, the quantity of annotated training data varies considerably between studies, making it difficult to obtain practical recommendations for new projects seeking to use normalization techniques.

Contributions This paper aims to provide the most comprehensive evaluation and analysis of historical text normalization systems so far. Motivated by a systematic review of previous work on this topic (Sec. 2), only publicly available normalization systems covering a wide range of proposed techniques are selected (Sec. 3) and evaluated across a diverse collection of historical datasets covering eight languages (Sec. 4). This is followed by a detailed analysis of the effect of training data quantity and a critical discussion of evaluation methods for assessing normalization quality (Sec. 5).

The datasets and code are made freely available whenever possible,[2] along with detailed instructions on how to reproduce the experiments.

[1] This work largely builds upon the author's doctoral thesis (Bollmann, 2018), the research for which was carried out at Ruhr-Universität Bochum, Germany.

[2] https://github.com/coastalcph/histnorm; one dataset could not be included due to licensing restrictions.

Proceedings of NAACL-HLT 2019, pages 3885–3898
Minneapolis, Minnesota, June 2 - June 7, 2019.

2 A Brief Survey of Automatic Historical Text Normalization

The following overview is broadly organized by categories that each represent a conceptually or methodically different approach.

2.1 Substitution Lists

The conceptually simplest form of normalization is to look up each historical variant in a precompiled list that maps it to its intended normalization. This approach can go by many names, such as *lexical substitution*, *dictionary lookup*, *wordlist mapping*, or *memorization*. While it does not generalize in any way to variants that are not covered by the list, it has proven highly effective as a component in several normalization systems, such as the semi-automatic VARD tool (Rayson et al., 2005; Baron and Rayson, 2008) or the fully automatic Norma tool (Bollmann, 2012).

2.2 Rule-based Methods

Rule-based approaches try to encode regularities in spelling variants—e.g., historical ⟨*v*⟩ often representing modern ⟨*u*⟩—in the form of replacement rules, typically including context information to discriminate between different usages of a character. Some of the earliest approaches to normalization are rule-based, with rules being created manually for one particular language, such as Old Icelandic (Fix, 1980) or Old German (Koller, 1983).

VARD 2 uses "letter replacement rules" to construct normalization candidates, but is not necessarily concerned with precision due to its interactive nature (Baron and Rayson, 2008). Bollmann et al. (2011) describe a supervised learning algorithm to automatically derive context-aware replacement rules from training data, including "identity rules" that leave a character unchanged, then apply one rule to each character of a historical word form to produce a normalization. Porta et al. (2013) model phonological sound change rules for Old Spanish using finite-state transducers; Etxeberria et al. (2016) describe a similarly motivated model that can be trained in a supervised manner.

Rule-based methods are also commonly found when the goal is not to produce a single best normalization, but to cluster a group of spelling variants (Giusti et al., 2007) or to retrieve occurrences of variant spellings given a modern form in an information retrieval (IR) scenario (Ernst-Gerlach and Fuhr, 2006; Koolen et al., 2006).

2.3 Distance-based Methods

Approaches using edit distance measures (such as Levenshtein distance; Levenshtein, 1966) are most commonly found in an IR context, since measures that compare two word forms are a natural fit for matching a search term with relevant word forms in a historical document (e.g., Robertson and Willett, 1993). Weighted variants of distance measures can be used to assign lower costs to more likely edit operations (Kempken et al., 2006; Hauser and Schulz, 2007).

In a normalization context, distance measures can be used to compare historical variants to entries in a contemporary full-form lexicon (Kestemont et al., 2010; Jurish, 2010a). Norma includes a distance-based component whose edit weights can be learned from a training set of normalizations (Bollmann, 2012). Pettersson et al. (2013a) find a similar approach to be more effective than hand-crafted rules on Swedish. Sometimes, the line between distance-based and rule-based methods get blurred; Adesam et al. (2012) use the Levenshtein algorithm to derive "substitution rules" from training data, which are then used to link up historical Swedish forms with lexicon entries; van Halteren and Rem (2013) describe a comparable approach for Dutch.

Furthermore, distance measures also lend themselves to unsupervised approaches for clustering historical variants of the same modern form, where identifying the precise modern form is not necessarily required (Amoia and Martínez, 2013; Barteld et al., 2015).

2.4 Statistical Models

In a probabilistic view of the normalization task, the goal is to optimize the probability $p(t|s)$ that a contemporary word form t is the normalization of a historical word form s. This can be seen as a *noisy channel model*, which has been used for normalization by, e.g., Oravecz et al. (2010) and Etxeberria et al. (2016).

More commonly, *character-based statistical machine translation (CSMT)* has been applied to the normalization task. Instead of translating a sentence as a sequence of tokens, these approaches "translate" a historical word form as a sequence of characters. This has been found to be very effective for a variety of historical languages, such as Spanish (Sánchez-Martínez et al., 2013), Icelandic and Swedish (Pettersson et al.,

2013b), Slovene (Scherrer and Erjavec, 2013, 2016; Ljubešić et al., 2016), as well as Hungarian, German, and English (Pettersson, 2016), where it is usually found to outperform previous approaches.

Pettersson et al. (2014) find that a CSMT system often performs best in a comparison with a filtering method and a distance-based approach on five different languages. Schneider et al. (2017) compare VARD 2 to CSMT on English and find that VARD 2 performs slightly better. Domingo and Casacuberta (2018) evaluate both word-based and character-based models and find that SMT outperforms a neural network model.

2.5 Neural Models

Neural network architectures have become popular for a variety of NLP tasks, and historical normalization is no exception. *Character-based neural machine translation (CNMT)* is the logical neural equivalent to the CSMT approach, and has first been used for normalization of historical German (Bollmann et al., 2017; Korchagina, 2017) using encoder–decoder models with long short-term memory (LSTM) units. Robertson and Goldwater (2018) present a more detailed evaluation of this architecture on five different languages. Hämäläinen et al. (2018) evaluate SMT, NMT, an edit-distance approach, and a rule-based finite-state transducer, and advocate for a combination of these approaches to make use of their individual strengths; however, they restrict their evaluation to English.

Other neural architectures have rarely been used for normalization so far. Al Azawi et al. (2013) and Bollmann and Søgaard (2016) frame the normalization task as a sequence labelling problem, labelling each character in the historical word form with its normalized equivalent. Kestemont et al. (2016) use convolutional networks for lemmatization of historical Dutch. Overall, though, the encoder–decoder model with recurrent layers is the dominant approach.

2.6 Beyond Token-Level Normalization

The presented literature almost exclusively focuses on models where the input is a single token. In theory, it would be desirable to include context from the surrounding tokens, as some historical spellings can have more than one modern equivalent depending on the context in which they are used (e.g., historical *ther* could represent *their* or *there*). Remarkably few studies have attempted this so far: Jurish (2010b) uses hidden Markov models to select between normalization candidates; Mitankin et al. (2014) use a language model in a similar vein; Ljubešić et al. (2016) experiment with "segment-level" input, i.e., a string of several historical tokens as input to a normalizer. Since this area is currently very underexplored, it warrants a deeper investigation that goes beyond the scope of this paper.

3 Experimental Setup

Systems The selection of normalization systems follows two goals: (i) to include at least one system for each major category as identified in Sec. 2; and (ii) to use only freely available tools in order to facilitate reproduction and application of the described methods. To that effect, this study compares the following approaches:

- **Norma**[3] (Bollmann, 2012), which combines substitution lists, a rule-based normalizer, and a distance-based algorithm, with the option of running them separately or combined. Importantly, it implements supervised learning algorithms for all of these components and is not restricted to a particular language.
- **cSMTiser**[4] (Ljubešić et al., 2016; Scherrer and Ljubešić, 2016), which implements a normalization pipeline using character-based statistical machine translation (CSMT) using the Moses toolkit (Koehn et al., 2007).
- **Neural machine translation (NMT)**, in the form of two publicly available implementations: (i) the model by Bollmann (2018), also used in Bollmann et al. (2018);[5] and (ii) the model by Tang et al. (2018).[6]

Two systems were chosen for the NMT approach as they use very different hyperparameters, despite both using comparable neural encoder–decoder models: Bollmann (2018) uses a single LSTM layer with dimensionality 300 in the encoder and decoder, while Tang et al. (2018) use six vanilla RNN cells with dimensionality 1024.

[3] https://github.com/comphist/norma

[4] https://github.com/clarinsi/csmtiser

[5] I reimplemented the model here using the XNMT toolkit (Neubig et al., 2018).

[6] https://github.com/tanggongbo/normalization-NMT; their model uses the deep transition architecture of Sennrich et al. (2017, Sec. 2.3.1) as implemented by Marian (Junczys-Dowmunt et al., 2018).

Dataset/Language		Time Period	Genre	Tokens		
				TRAIN	DEV	TEST
DE_A	German (Anselm)	14th–16th c.	Religious	233,947	45,996	45,999
DE_R	German (RIDGES)	1482–1652	Science	41,857	9,712	9,587
EN	English	1386–1698	Letters	147,826	16,334	17,644
ES	Spanish	15th–19th c.	Letters	97,320	11,650	12,479
HU	Hungarian	1440–1541	Religious	134,028	16,707	16,779
IS	Icelandic	15th c.	Religious	49,633	6,109	6,037
PT	Portuguese	15th–19th c.	Letters	222,525	26,749	27,078
SL_B	Slovene (Bohorič)	1750–1840s	Diverse	50,023	5,841	5,969
SL_G	Slovene (Gaj)	1840s–1899	Diverse	161,211	20,878	21,493
SV	Swedish	1527–1812	Diverse	24,458	2,245	29,184

Table 1: Historical datasets used in the experiments

Datasets Table 1 gives an overview of the historical datasets. They are taken from Bollmann (2018) and represent the largest and most varied collection of datasets used for historical text normalization so far, covering eight languages from different language families—English, German, Hungarian, Icelandic, Spanish, Portuguese, Slovene, and Swedish—as well as different text genres and time periods. Furthermore, most of these have also been used in previous work, such as the English, Hungarian, Icelandic, and Swedish datasets (e.g., Pettersson et al., 2014; Pettersson, 2016; Robertson and Goldwater, 2018; Tang et al., 2018) and the Slovene datasets (e.g., Ljubešić et al., 2016; Scherrer and Erjavec, 2016; Etxeberria et al., 2016; Domingo and Casacuberta, 2018).

Additionally, contemporary datasets are required for the rule-based and distance-based components of Norma, as they expect a list of valid target word forms to function properly. For this, we want to choose resources that are readily available for many languages and are reliable, i.e., consist of carefully edited text. Here, I choose a combination of three sources:[7] (i) the normalizations in the training sets, (ii) the Europarl corpus (Koehn, 2005), and (iii) the parallel Bible corpus by Christodouloupoulos and Steedman (2015). The only exception is Icelandic, which is not covered by Europarl; here, we can follow Pettersson (2016) instead by using data from two specialized resources, the BÍN database (Bjarnadóttir, 2012) and the MÍM corpus (Helgadóttir et al., 2012). This way, we obtain full-form lexica of 12k–64k word types from the Bible corpus, 55k–268k types from Europarl, and 2.8M types from the Icelandic resources.

Preprocessing The most important preprocessing decisions[8] are (i) to lowercase all characters and (ii) to remove all punctuation-only tokens. Both capitalization and punctuation often cannot be handled correctly without considering token context, which all current normalization models do not do. Furthermore, their usage can be very erratic in historical texts, potentially distorting the evaluation; e.g., when a text uses punctuation marks according to modern conventions, their normalization is usually trivial, resulting in artificial gains in normalization accuracy that other texts do not get. At the same time, most previous work has not followed these same preprocessing guidelines, making a direct comparison more difficult. This work tries to make up for this by evaluating many different systems, effectively reproducing some of these previous results instead.

4 Evaluation

All models are trained and evaluated separately for each dataset by calculating word accuracy over all tokens. In particular, there is no step to discriminate between tokens that require normalization and those that do not; all word forms in the datasets are treated equally.

For Norma, all components are evaluated both separately and combined, as the former gives us insight into the performance of each individual

[7] Detailed descriptions of the data extraction procedure can be found in the Supplementary Material.

[8] The full preprocessing steps can be found in the Supplementary Material.

Method	Dataset									
	DE_A	DE_R	EN	ES	HU	IS	PT	SL_B	SL_G	SV
Identity	*30.63*	*44.36*	*75.29*	*73.40*	*17.53*	*47.62*	*65.19*	*40.74*	*85.38*	*58.59*
Maximum	*94.64*	*96.46*	*98.57*	*97.40*	*98.70*	*93.46*	*97.65*	*98.71*	*98.96*	*98.97*
Norma, Lookup	83.86	82.15	92.45	92.51	74.58	82.84	91.67	81.76	93.90	83.80
Norma, Rule-based	76.48	82.52	90.85	88.59	78.73	83.72	86.33	86.09	91.63	85.23
Norma, Distance-based	58.92	73.30	83.92	84.41	62.38	69.95	77.28	71.02	88.20	76.03
Norma (Combined)	88.02	86.55	94.60	94.41	86.83	*86.85	94.19	89.45	91.44	87.12
cSMTiser	88.82	*88.06	*95.21	*95.01	*91.63	*87.10	*95.09	*93.18	*95.99	**91.13**
$cSMTiser_{+LM}$	86.69	*88.19	**95.24**	**95.02**	**91.70**	*86.83	**95.18**	**93.30**	**96.01**	*91.11
NMT (Bollmann, 2018)	89.16	*88.07	94.80	*94.83	91.17	86.45	94.64	91.61	95.19	90.27
NMT (Tang et al., 2018)	**89.64**	**88.22**	94.95	*94.84	*91.65	**87.31**	94.51	92.60	*95.85	90.39
†SMT (Pettersson et al., 2014)	–	–	94.3–	–	80.1–	71.8–	–	–	–	92.9–
†NMT (Tang et al., 2018)	–	–	94.69	–	91.69	87.59	–	–	–	91.56

Table 2: Word accuracy of different normalization methods on the test sets of the historical datasets, in percent; best result for each dataset in bold; results marked with an asterisk (*) are not significantly different from the best result using McNemar's test at $p < 0.05$. † indicates scores that were not (re)produced here, but reported in previous work; they might not be strictly comparable due to differences in data preprocessing (cf. Sec. 3). Additionally, *Identity* shows the accuracy when leaving all word forms unchanged, while *Maximum* gives the theoretical maximum accuracy with purely token-level methods.

component, while the latter is reported to produce the best results (Bollmann, 2012). For cSMTiser, the authors suggest using additional monolingual data to improve the language model; the contemporary datasets are used for this purpose and the model is trained both without and with this additional data; the latter is denoted $cSMTiser_{+LM}$. For NMT, the model by Bollmann (2018) is evaluated using an ensemble of five models; the model by Tang et al. (2018) is trained on character-level input using the default settings provided by their implementation.[9]

To illustrate how challenging the normalization task is on different datasets, we can additionally look at the *identity baseline*—i.e., the percentage of tokens that do *not* need to be normalized—as well as the *maximum accuracy* obtainable if each word type was mapped to its most frequently occurring normalization. The latter gives an indication of the extent of ambiguity in the datasets and the disadvantage of not considering token context (cf. Sec. 2.6).

Results Table 2 shows the results of this evaluation. The extent of spelling variation varies greatly between datasets, with less than 15% of tokens requiring normalization (SL_G) to more than 80% (HU). The maximum accuracy is above 97% for most datasets, suggesting that we can obtain high normalization accuracy in principle even without considering token context.

For the normalization systems, we observe significantly better word accuracy with SMT than NMT on four of the datasets, and non-significant differences on five others. There is only one dataset (DE_A) where the NMT system by Tang et al. (2018) gets significantly better word accuracy than other systems. This somewhat contradicts the results from Tang et al. (2018), who find NMT to usually outperform the SMT baseline by Pettersson et al. (2014). However, note that the results for the cSMTiser system are often significantly better than reported in previous work: e.g., on Hungarian, cSMTiser obtains 91.7% accuracy, but only 80.1% with the SMT system from Pettersson et al. (2014).

Overall, the deep NMT model by Tang et al. (2018) consistently outperforms the shallow one by Bollmann (2018). cSMTiser seems to benefit from the added contemporary data for language modelling, though the effect is not significant on any individual dataset. Finally, while Norma does produce competitive results on sev-

[9]This is the "Att-RNN" setting reported in their paper; due to the high computational demands of the model, it was not feasible to run experiments with multiple configurations.

Method	Dataset									
	DE_A	DE_R	EN	ES	HU	IS	PT	SL_B	SL_G	SV
Norma, Lookup	0.41	0.31	**0.38**	**0.35**	0.43	**0.38**	**0.39**	**0.44**	0.44	0.29
Norma, Rule-based	0.40	0.33	0.43	0.39	0.38	0.40	0.45	0.47	0.46	0.32
Norma, Distance-based	0.42	0.34	0.46	0.44	0.41	0.44	0.50	0.52	**0.39**	0.38
Norma (Combined)	0.41	0.33	0.45	0.42	0.34	0.42	0.51	0.51	0.42	0.31
cSMTiser	**0.37**	**0.26**	0.39	0.41	**0.26**	0.40	0.50	0.53	0.56	0.24
$cSMTiser_{+LM}$	0.39	0.27	0.39	0.42	0.27	0.41	0.50	0.53	0.56	0.24
NMT (Bollmann, 2018)	0.38	**0.26**	0.39	0.43	0.27	0.40	0.48	0.47	0.51	**0.23**
NMT (Tang et al., 2018)	0.38	0.27	**0.38**	0.42	**0.26**	0.41	0.46	0.50	0.56	0.24

(a) CER_I: character error rate on the subset of incorrect normalizations (lower is better)

Norma, Lookup	8.47	16.72	8.33	27.81	2.86	–	4.57	–	–	7.42
Norma, Rule-based	12.82	26.73	8.48	26.33	12.44	–	5.65	–	–	15.38
Norma, Distance-based	7.71	19.10	7.26	28.62	10.93	–	5.58	–	–	12.18
Norma (Combined)	16.97	28.94	**9.86**	43.55	20.00	–	**11.13**	–	–	**20.75**
cSMTiser	**17.92**	**34.67**	8.27	**43.82**	**20.44**	–	6.09	–	–	17.73
$cSMTiser_{+LM}$	12.23	33.83	8.46	42.93	20.40	–	6.60	–	–	17.58
NMT (Bollmann, 2018)	17.24	33.65	7.96	39.22	18.30	–	5.92	–	–	16.65
NMT (Tang et al., 2018)	16.34	34.19	9.43	40.99	19.84	–	6.46	–	–	18.93

(b) Stemming accuracy: percentage of incorrect normalizations with correct word stems (higher is better)

Table 3: Evaluations on the subset of incorrect normalizations only; best results for each dataset in bold. Note that this subset is different for each system, so for comparisons between systems, these numbers should be considered in conjunction with word accuracy scores from Table 2.

eral datasets (particularly in the "combined" setting), it is generally significantly behind the SMT and NMT methods.

5 Analysis

5.1 Measuring Normalization Quality

While word accuracy is easily interpretable, it is also a very crude measure, as it classifies predictions as correct/incorrect without considering the type of error(s) made by the model. *Character error rate (CER)* has sometimes been suggested as a complement to address this issue, but I believe this is not very insightful: For any normalization system that achieves a reasonably high word accuracy, CER will highly correlate with accuracy simply because CER equals zero for any word that is accurately normalized.[10] At the same time, there is a need for a more fine-grained way to assess the normalization quality. Consider the following example from the Hungarian dataset with its predicted normalization from the NMT system by Bollmann (2018):

(1) ORIG yduewzewlendewk
GOLD üdvözülendőek
PRED üdvözülendők

Here, the prediction matches the correct target form almost perfectly, but would be counted as incorrect since it misses an insertion of the letter ⟨*e*⟩ towards the end. In this vein, it will be treated the same by the word accuracy measure as a prediction that, e.g., had left the original form unchanged.

CER_I One alternative is to consider character error rate *on the subset of incorrect normalizations only.* This way, CER becomes a true complement to word accuracy by assessing the *magnitude of error* that a normalization model makes when it is not perfectly accurate. The results of this measure, denoted CER_I, are shown in Table 3a. The lowest CER_I score is often achieved

[10] When comparing word accuracy scores in Table 2 with the same configurations evaluated using CER, they correlate with Pearson's $r \approx -0.96$.

by Norma's lookup module, which leaves historical word forms unchanged if they are not in its lookup wordlist learned during training. This suggests that the incorrect predictions made by other systems are often *worse* than just leaving the historical spelling unchanged.

Stemming Another problem of CER is that all types of errors are treated the same: a one-letter difference in inflection, such as *king – kings* or *came – come*, would be treated identically to an error that changes the meaning of the word (*bids – beds*) or results in a non-word (*creature – cryature*). I propose an approach that, to the best of my knowledge, has not been used in normalization evaluation before: measure accuracy on *word stems,* i.e., process both the reference normalization and the prediction with an automatic stemming algorithm and check if both stems match. For this evaluation, I choose the Snowball stemmer (Porter, 2001) as it contains stemming algorithms for many languages (including the ones represented here except for Icelandic and Slovene) and is publicly available.[11]

Table 3b shows the accuracy on word stems, again only evaluated on the subset of incorrect normalizations, as this better highlights the differences between settings. This evaluation reveals some notable differences between *datasets*: For example, while the English and Spanish datasets have very comparable accuracy scores overall (cf. Tab. 2), they show very different characteristics in the stemming evaluation; for English, only up to 9.86% of incorrect predictions show the correct word stem, while for Spanish the number is up to 43.82%. Examining predictions on the dev set, many of the incorrectly predicted cases in Spanish result from mistakes in placement of diacritics, such as *ésta – está* or *envíe – envié*; the stemming algorithm removes diacritics and can therefore match these instances. Overall, this gives an indication that the errors made on the Spanish dataset are less severe than those on English, despite comparable word accuracy scores and a usually higher CER_I for Spanish.

This case study shows that stemming can be a useful tool for error analysis in normalization models and reveal characteristics that neither word accuracy nor CER alone can show.

[11] http://snowballstem.org/

5.2 Effect of Training Data Quantity

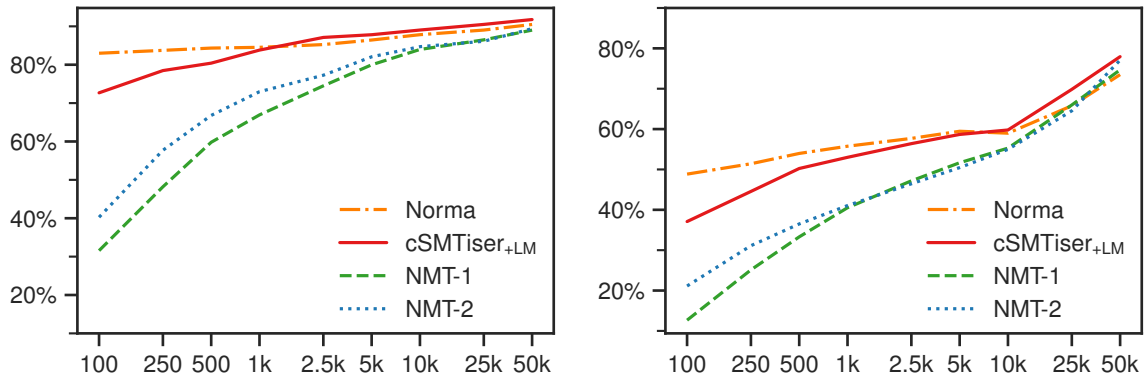
Supervised methods for historical text normalization have been evaluated with highly varying amounts of training data: e.g., Domingo and Casacuberta (2018) train a normalizer for 17th century Spanish on 436k tokens; Etxeberria et al. (2016) use only 8k tokens to train a normalizer for Basque. Even in the evaluation in Sec. 4, training set sizes varied between 24k and 234k tokens, depending on the dataset. Furthermore, many research projects seeking to use automatic normalization techniques cannot afford to produce training data in high quantity. All of this raises the question how different normalization systems perform with varying amounts of training data, and whether reasonable normalization results can be achieved in a low-resource scenario.

Methodology All models are retrained on varying subsets of the training data, with sizes ranging from 100 tokens to 50,000 tokens. However, the lower the training set size is, the higher the potential variance when training on it, since random factors such as the covered spelling variants or vocabulary are more likely to impact the results. Therefore, I choose the following approach: For each dataset and training size, up to ten different training splits are extracted,[12] and a separate model is trained on each one. Each model is then evaluated on the respective development dataset, and only the average accuracy across all splits is considered.

Results Figure 2 shows two learning curves that are representative for most of the datasets.[13] They reveal that Norma (in the "combined" setting) performs best in extremely low-resource scenarios, but is overtaken by the SMT approach as more training data becomes available; usually already around 500–1000 tokens. The NMT models have a steeper learning curve, needing more training data to become competitive. Extrapolating this trend, it is conceivable that the NMT models would simply need more training data than our current datasets provide in order to consistently outperform the SMT approach. On the other hand, there appears to be no correlation between the size of the training set (cf. Tab. 1) and the relative per-

[12] The ten training splits consist of chunks of n tokens that are spaced equidistantly across the full training set; for larger n, the number of chunks is reduced so that no splits overlap to more than 50%.

[13] Plots for all datasets can be found in the Appendix.



(a) English

(b) Hungarian

Figure 2: Word accuracy on the development sets for different amounts of training data (note that the x-axis is log-scaled); NMT-1 is the model by Bollmann (2018), NMT-2 is the model by Tang et al. (2018).

formance of NMT vs. SMT (cf. Tab. 2) in the experiments. Since I am not aware of larger datasets for the historical normalization task, this remains an open question for now.

A remarkable result is that very small amounts of training data can already be helpful for the normalization task. The English dataset has comparatively little spelling variation to begin with: leaving all words unnormalized already results in an accuracy of 75.5%. Still, with as little as 100 tokens for training, applying the Norma tool raises the accuracy above 83%. For Hungarian, the same amount of training data raises the accuracy from 17.8% (unnormalized) to around 50%. It would be interesting to further compare these results with fully unsupervised methods.

5.3 Out-of-Vocabulary Words

Robertson and Goldwater (2018) highlight the importance of evaluating separately on seen vs. unseen tokens, i.e., tokens that have also been in the training set (in-vocabulary) and those that have not (out-of-vocabulary), as well as comparing to a naive memorization baseline. These numbers are presented in Table 4. For unseen tokens (Tab. 4b), the accuracy scores follow generally the same trend as in the full evaluation of Tab. 2; i.e., SMT performs best in most cases. For seen tokens (Tab. 4a), however, Norma's lookup component—which implements naive memorization—obtains the highest score on nine datasets.

These observations suggest a new normalization strategy: apply the naive lookup on the subset of in-vocabulary tokens and the SMT/NMT models on the subset of out-of-vocabulary tokens only. Table 5 shows the results of this strategy.[14] On nine datasets, it performs better than always using the learned models (as in Tab. 2), and this difference is statistically significant on five of them. These results support the claim from Robertson and Goldwater (2018) that "learned models should typically only be applied to *unseen* tokens."

6 Conclusion

This paper presented a large study of historical text normalization. Starting with a systematic survey of the existing literature, four different systems (based on supervised learning) were evaluated and compared on datasets from eight different languages. On the basis of these results, we can extract some practical recommendations for projects seeking to employ normalization techniques:

1. to use the Norma tool when only little training data (<500 tokens) is available;
2. to use cSMTiser otherwise, ideally with additional data for language modelling; and
3. to make use of the naive memorization/lookup technique for in-vocabulary tokens when possible.

Furthermore, the qualitative analysis (in Sec. 5.1) should encourage authors evaluating normalization systems to use task-motivated approaches, such as evaluation on word stems, to provide

[14] The non-lookup components of Norma are not included in this evaluation since "Norma (Combined)" effectively implements such a strategy already.

Method	Dataset									
	DE_A	DE_R	EN	ES	HU	IS	PT	SL_B	SL_G	SV
Norma, Lookup/Combined	**92.36**	**93.66**	**97.46**	**96.59**	**96.81**	**89.51**	**97.04**	**97.15**	**98.17**	97.61
Norma, Rule-based	80.34	89.26	93.17	89.98	88.42	86.21	88.77	93.61	96.10	92.10
Norma, Distance-based	60.79	78.85	86.77	85.89	67.04	70.99	78.93	77.49	94.54	84.17
cSMTiser	92.18	93.25	97.10	96.33	96.33	89.27	96.80	96.73	98.09	**97.66**
$cSMTiser_{+LM}$	90.52	93.45	97.15	96.42	96.33	88.99	96.82	96.90	98.07	**97.66**
NMT (Bollmann, 2018)	91.91	93.29	97.19	96.37	96.18	88.67	96.72	95.92	97.56	97.01
NMT (Tang et al., 2018)	92.25	93.41	97.19	96.27	96.43	89.34	96.60	96.84	97.89	96.90

(a) In-vocabulary/seen tokens

Norma, Lookup	3.91	19.92	30.42	46.72	3.28	29.40	28.25	18.07	68.07	39.85
Norma, Rule-based	40.27	46.06	62.06	72.97	47.66	63.73	57.50	54.99	64.59	63.37
Norma, Distance-based	41.33	43.25	48.63	67.78	47.43	61.64	57.79	44.23	49.82	50.15
Norma (Combined)	47.25	48.13	59.18	69.93	54.83	65.52	60.62	57.57	50.71	53.73
cSMTiser	57.25	59.96	**71.78**	**80.22**	76.59	69.70	74.69	**78.49**	83.30	**70.35**
$cSMTiser_{+LM}$	50.69	59.76	71.70	79.24	**76.86**	69.55	**75.91**	78.40	**83.56**	70.26
NMT (Bollmann, 2018)	63.24	59.83	65.17	77.57	75.13	68.66	70.00	73.75	80.87	68.78
NMT (Tang et al., 2018)	**65.09**	**60.16**	67.15	78.75	76.33	**71.04**	69.90	75.04	83.46	69.66

(b) Out-of-vocabulary/unseen tokens

Table 4: Word accuracy for seen/unseen tokens separately (cf. Sec. 5.3); best results for each dataset in bold.

Method	Dataset									
	DE_A	DE_R	EN	ES	HU	IS	PT	SL_B	SL_G	SV
Best without lookup	**89.64*	**88.22*	*95.24*	*95.02*	*91.70*	**87.31*	*95.18*	*93.30*	**96.01*	***91.13***
cSMTiser	88.98	*88.41	**95.54**	**95.25**	*92.00	*87.31	*95.31	**93.52**	*96.06	*91.09
$cSMTiser_{+LM}$	88.35	*88.37	*95.53	*95.17	**92.07**	*87.30	**95.39**	*93.50	**96.10**	*91.07
NMT (Bollmann, 2018)	89.56	*88.38	95.05	95.03	91.66	*87.20	94.93	92.60	95.71	90.72
NMT (Tang et al., 2018)	**89.74**	**88.45**	95.19	*95.13	*91.94	**87.46**	94.92	92.85	*96.08	*90.93

Table 5: Word accuracy for the "lookup on *seen* tokens, learned models on *unseen* tokens" strategy, following Robertson and Goldwater (2018) (cf. Sec. 5.3), compared to the best result without this strategy (according to Table 2). Best result for each dataset in bold; results marked with an asterisk (*) are not significantly different from the best result using McNemar's test at $p < 0.05$.

deeper insight into the properties of their models and datasets.

Detailed information on how to train and apply all of the evaluated techniques is made available online at `https://github.com/coastalcph/histnorm`.

Acknowledgments

I would like to thank my PhD supervisor, Stefanie Dipper, for her continuous support over many years that culminated in the doctoral thesis on which this paper is based; the acknowledgments in that thesis largely extend to this paper as well. Many thanks to Anders Søgaard for many helpful discussions and for supporting the follow-up experiments conducted for this paper. Further thanks go to the anonymous reviewers whose helpful suggestions have largely been incorporated here.

I gratefully acknowledge the donation of a Titan Xp GPU by the NVIDIA Corporation that was used for a substantial part of this research.

References

Yvonne Adesam, Malin Ahlberg, and Gerlof Bouma. 2012. *bokstaffua, bokstaffwa, bokstafwa, bokstaua, bokstawa...* Towards lexical link-up for a corpus of Old Swedish. In *Proceedings of the 11th Conference on Natural Language Processing (KONVENS 2012), LThist 2012 workshop*, pages 365–369, Vienna, Austria.

Mayce Al Azawi, Muhammad Zeshan Afzal, and Thomas M. Breuel. 2013. Normalizing historical orthography for OCR historical documents using LSTM. In *Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing (HIP '13)*, pages 80–85, Washington, DC.

Marilisa Amoia and José Manuel Martínez. 2013. Using comparable collections of historical texts for building a diachronic dictionary for spelling normalization. In *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 84–89. Association for Computational Linguistics.

Alistair Baron and Paul Rayson. 2008. VARD 2: A tool for dealing with spelling variation in historical corpora. In *Proceedings of the Postgraduate Conference in Corpus Linguistics*.

Fabian Barteld, Ingrid Schröder, and Heike Zinsmeister. 2015. Unsupervised regularization of historical texts for POS tagging. In *Proceedings of the Workshop on Corpus-Based Research in the Humanities (CRH)*, pages 3–12, Warsaw, Poland.

Kristín Bjarnadóttir. 2012. The database of modern Icelandic inflection (Beygingarlýsing íslensks nútímamáls). In *Proceedings of the Workshop on Language Technology for Normalisation of Less-Resourced Languages*, pages 13–18.

Marcel Bollmann. 2012. (Semi-)automatic normalization of historical texts using distance measures and the Norma tool. In *Proceedings of the Second Workshop on Annotation of Corpora for Research in the Humanities (ACRH-2)*, pages 3–12, Lisbon, Portugal.

Marcel Bollmann. 2018. Normalization of historical texts with neural network models. *Bochumer Linguistische Arbeitsberichte*, 22.

Marcel Bollmann, Joachim Bingel, and Anders Søgaard. 2017. Learning attention for historical text normalization by learning to pronounce. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 332–344. Association for Computational Linguistics.

Marcel Bollmann, Florian Petran, and Stefanie Dipper. 2011. Rule-based normalization of historical texts. In *Proceedings of the International Workshop on Language Technologies for Digital Humanities and Cultural Heritage*, pages 34–42, Hissar, Bulgaria.

Marcel Bollmann and Anders Søgaard. 2016. Improving historical spelling normalization with bi-directional LSTMs and multi-task learning. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 131–139. The COLING 2016 Organizing Committee.

Marcel Bollmann, Anders Søgaard, and Joachim Bingel. 2018. Multi-task learning for historical text normalization: Size matters. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, pages 19–24. Association for Computational Linguistics.

Christos Christodouloupoulos and Mark Steedman. 2015. A massively parallel corpus: the Bible in 100 languages. *Language Resources and Evaluation*, 49(2):375–395.

Miguel Domingo and Francisco Casacuberta. 2018. Spelling normalization of historical documents by using a machine translation approach. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 129–137.

Andrea Ernst-Gerlach and Norbert Fuhr. 2006. Generating search term variants for text collections with historic spellings. In *Proceedings of the 28th European Conference on Information Retrieval Research (ECIR 2006)*, Lecture Notes in Computer Science, pages 49–60, Berlin. Springer.

Izaskun Etxeberria, Iñaki Alegria, Larraitz Uria, and Mans Hulden. 2016. Evaluating the noisy channel model for the normalization of historical texts: Basque, Spanish and Slovene. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1064–1069, Paris, France. European Language Resources Association (ELRA).

Hans Fix. 1980. Automatische Normalisierung – Vorarbeit zur Lemmatisierung eines diplomatischen altisländischen Textes. In Paul Sappler and Erich Straßner, editors, *Maschinelle Verarbeitung altdeutscher Texte. Beiträge zum dritten Symposion, Tübingen 17.–19. Februar 1977*, pages 92–100. Niemeyer, Tübingen.

Rafael Giusti, Arnaldo Candido Jr, Marcelo Muniz, Lívia Cucatto, and Sandra Aluísio. 2007. Automatic detection of spelling variation in historical corpus: An application to build a Brazilian Portuguese spelling variants dictionary. In *Proceedings of the Corpus Linguistics Conference (CL2007)*, Birmingham, UK.

Hans van Halteren and Margit Rem. 2013. Dealing with orthographic variation in a tagger-lemmatizer for fourteenth century Dutch charters. *Language Resources and Evaluation*, 47(4):1233–1259.

Mika Hämäläinen, Tanja Säily, Jack Rueter, Jörg Tiedemann, and Eetu Mäkelä. 2018. Normalizing Early English letters to present-day English spelling. In *Proceedings of the Second Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 87–96. Association for Computational Linguistics.

Andreas W. Hauser and Klaus U. Schulz. 2007. Unsupervised learning of edit distance weights for retrieving historical spelling variations. In *Proceedings of the First Workshop on Finite-State Techniques and Approximate Search (FSTAS 2007)*, pages 1–6, Borovets, Bulgaria.

Sigrún Helgadóttir, Ásta Svavarsdóttir, Eiríkur Rögnvaldsson, Kristín Bjarnadóttir, and Hrafn Loftsson. 2012. The tagged Icelandic corpus (MíM). In *Proceedings of the Workshop on Language Technology for Normalisation of Less-Resourced Languages*, pages 67–72.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Bryan Jurish. 2010a. Comparing canonicalizations of historical German text. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 72–77, Uppsala, Sweden. Association for Computational Linguistics.

Bryan Jurish. 2010b. More than words: using token context to improve canonicalization of historical German. *Journal for Language Technology and Computational Linguistics*, 25(1):23–39.

Sebastian Kempken, Wolfram Luther, and Thomas Pilz. 2006. Comparison of distance measures for historical spelling variants. In Max Bramer, editor, *Artificial Intelligence in Theory and Practice*, pages 295–304. Springer, Boston, MA.

Mike Kestemont, Walter Daelemans, and Guy De Pauw. 2010. Weigh your words—memory-based lemmatization for Middle Dutch. *Literary and Linguistic Computing*, 25(3):287–301.

Mike Kestemont, Guy de Pauw, Renske van Nie, and Walter Daelemans. 2016. Lemmatization for variation-rich languages using deep learning. *Digital Scholarship in the Humanities*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Mercello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the ACL 2007 Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Gerhard Koller. 1983. Ein maschinelles Verfahren zur Normalisierung altdeutscher Texte. In Dietmar Peschel, editor, *Germanistik in Erlangen*, volume 31 of *Erlanger Forschungen*, pages 611–620. Universitätsbund Erlangen-Nürnberg, Erlangen.

Marijn Koolen, Frans Adriaans, Jaap Kamps, and Maarten de Rijke. 2006. A cross-language approach to historic document retrieval. In *Proceedings of the 28th European Conference on Information Retrieval Research (ECIR 2006)*, Lecture Notes in Computer Science, pages 407–419, Berlin. Springer.

Natalia Korchagina. 2017. Normalizing medieval German texts: from rules to deep learning. In *Proceedings of the NoDaLiDa 2017 Workshop on Processing Historical Language*, pages 12–17. Linköping University Electronic Press.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Nikola Ljubešić, Katja Zupan, Darja Fišer, and Tomaž Erjavec. 2016. Normalising Slovene data: historical texts vs. user-generated content. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*, volume 16 of *Bochumer Linguistische Arbeitsberichte*, pages 146–155, Bochum, Germany.

Petar Mitankin, Stefan Gerdjikov, and Stoyan Mihov. 2014. An approach to unsupervised historical text normalisation. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage (DATeCH '14)*, Madrid, Spain.

Graham Neubig, Matthias Sperber, Xinyi Wang, Matthieu Felix, Austin Matthews, Sarguna Padmanabhan, Ye Qi, Devendra Singh Sachan, Philip Arthur, Pierre Godard, John Hewitt, Rachid Riad, and Liming Wang. 2018. XNMT: The extensible neural machine translation toolkit. In *Conference of the Association for Machine Translation in the Americas (AMTA) Open Source Software Showcase*, Boston.

Csaba Oravecz, Bálint Sass, and Eszter Simon. 2010. Semi-automatic normalization of Old Hungarian codices. In *Proceedings of the ECAI 2010 Workshop on Language Technology for Cultural Heritage*, pages 55–59.

Eva Pettersson. 2016. *Spelling Normalisation and Linguistic Analysis of Historical Text for Information*

Extraction. Doctoral dissertation, Uppsala University, Department of Linguistics and Philology, Uppsala.

Eva Pettersson, Beáta Megyesi, and Joakim Nivre. 2013a. Normalisation of historical text using context-sensitive weighted Levenshtein distance and compound splitting. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)*, pages 163–179. Linköping University Electronic Press.

Eva Pettersson, Beáta Megyesi, and Joakim Nivre. 2014. A multilingual evaluation of three spelling normalisation methods for historical text. In *Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 32–41. Association for Computational Linguistics.

Eva Pettersson, Beáta Megyesi, and Jörg Tiedemann. 2013b. An SMT approach to automatic annotation of historical text. In *Proceedings of the Workshop on Computational Historical Linguistics at NODALIDA 2013*, NEALT Proceedings Series 18/Linköping Electronic Conference Proceedings 87, pages 54–69. Linköping University Electronic Press.

Jordi Porta, José-Luis Sancho, and Javier Gómez. 2013. Edit transducers for spelling variation in Old Spanish. In *Proceedings of the Workshop on Computational Historical Linguistics at NODALIDA 2013*, NEALT Proceedings Series 18/Linköping Electronic Conference Proceedings 87, pages 70–79. Linköping University Electronic Press.

Martin Porter. 2001. Snowball: A language for stemming algorithms.

Paul Rayson, Dawn Archer, and Nicholas Smith. 2005. VARD versus Word: A comparison of the UCREL variant detector and modern spell checkers on English historical corpora. In *Proceedings of the Corpus Linguistics Conference CL2005*, Birmingham, UK. University of Birmingham.

Alexander Robertson and Sharon Goldwater. 2018. Evaluating historical text normalization systems: How well do they generalize? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 720–725. Association for Computational Linguistics.

Alexander M. Robertson and Peter Willett. 1993. A comparison of spelling-correction methods for the identification of word forms in historical text databases. *Literary and Linguistic Computing*, 8(3):143–152.

Felipe Sánchez-Martínez, Isabel Martínez-Sempere, Xavier Ivars-Ribes, and Rafael C. Carrasco. 2013. An open diachronic corpus of historical Spanish: annotation criteria and automatic modernisation of spelling. *CoRR*, abs/1306.3692.

Yves Scherrer and Tomaž Erjavec. 2013. Modernizing historical slovene words with character-based SMT. In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, pages 58–62. Association for Computational Linguistics.

Yves Scherrer and Tomaž Erjavec. 2016. Modernising historical Slovene words. *Natural Language Engineering*, 22(6):881–905.

Yves Scherrer and Nikola Ljubešić. 2016. Automatic normalisation of the Swiss German ArchiMob corpus using character-level machine translation. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*, volume 16 of *Bochumer Linguistische Arbeitsberichte*, pages 248–255, Bochum, Germany.

Gerold Schneider, Eva Pettersson, and Michael Percillier. 2017. Comparing rule-based and SMT-based spelling normalisation for English historical texts. In *Proceedings of the NoDaLiDa 2017 Workshop on Processing Historical Language*, pages 40–46. Linköping University Electronic Press.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The University of Edinburgh's neural MT systems for WMT17. In *Proceedings of the Second Conference on Machine Translation*, pages 389–399. Association for Computational Linguistics.

Gongbo Tang, Fabienne Cap, Eva Pettersson, and Joakim Nivre. 2018. An evaluation of neural machine translation models on historical spelling normalization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1320–1331. Association for Computational Linguistics.

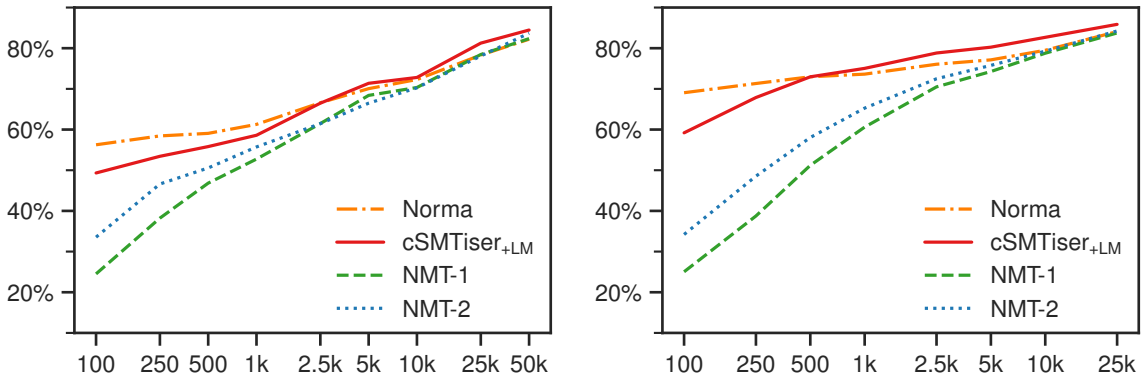
A Appendix

Figures 3 and 4 show plots of the learning curves for all of the datasets.

A.1 Preprocessing

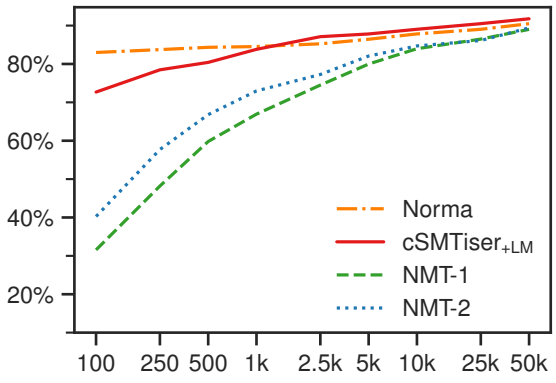
The full preprocessing steps of all datasets (both historical and contemporary) comprise of:

1. lowercasing all tokens;
2. filtering out pairs where either the historical token or the reference normalization is empty;
3. filtering out pairs where either the historical token or the reference normalization consists *only* of punctuation marks, defined as characters that belong to one of the Unicode "Punctuation" categories;

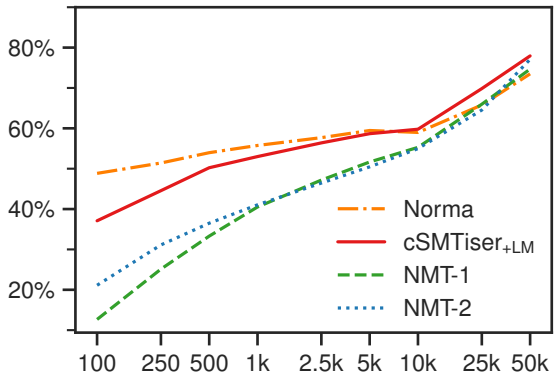


(a) German (Anselm)

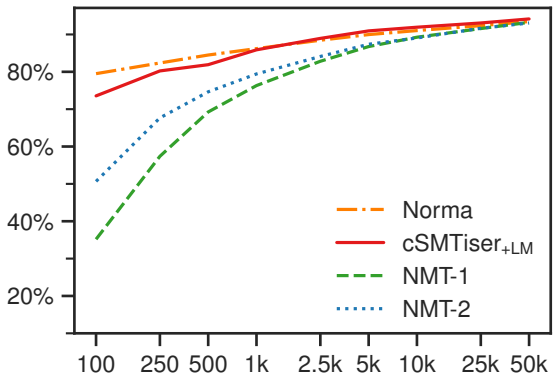
(b) German (RIDGES)



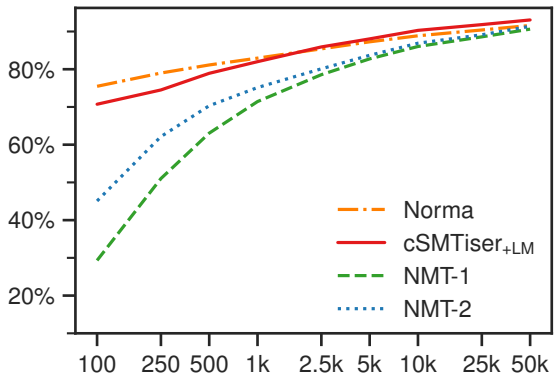
(c) English



(d) Hungarian

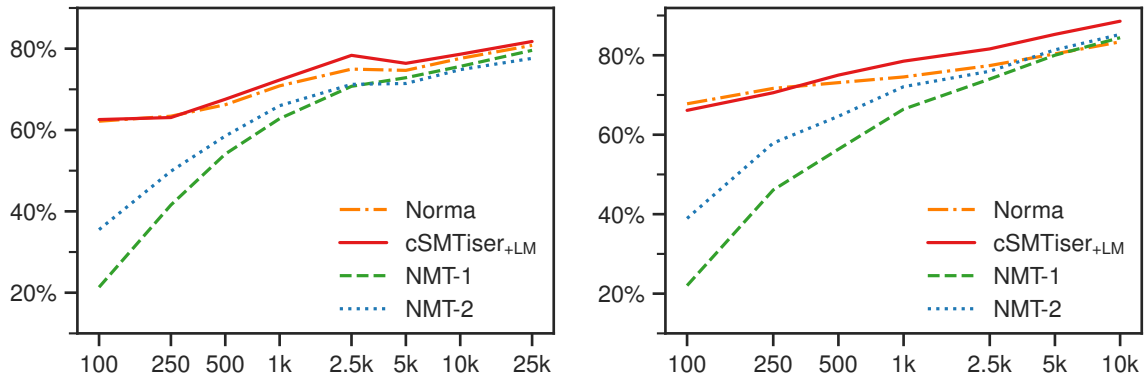


(e) Spanish



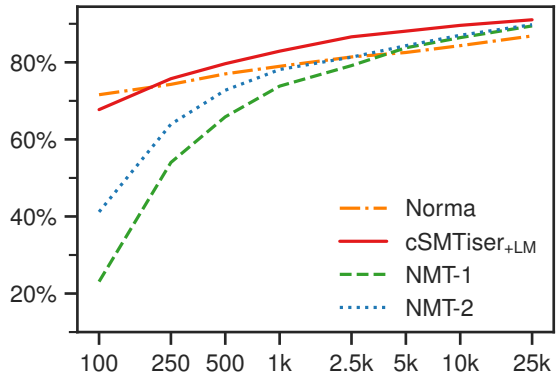
(f) Portuguese

Figure 3: Word accuracy on the development sets for different amounts of training data (note that the x-axis is log-scaled); NMT-1 is the model by Bollmann (2018), NMT-2 is the model by Tang et al. (2018).

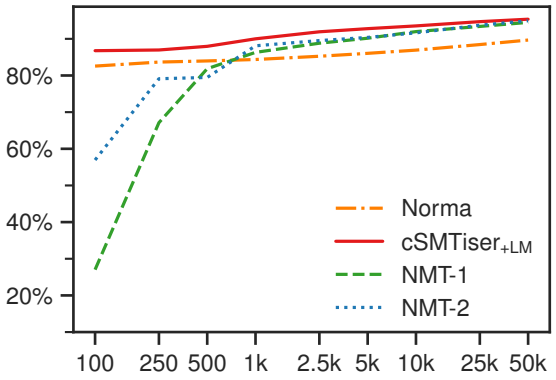


(a) Icelandic

(b) Swedish



(c) Slovene (Bohorič)



(d) Slovene (Gaj)

Figure 4: Word accuracy on the development sets (continued); NMT-1 is the model by Bollmann (2018), NMT-2 is the model by Tang et al. (2018).

4. replacing all digits with zeroes *iff* the digits in the historical token and the reference normalization match;

5. replacing actual space characters in either the historical token or the reference normalization with a special symbol that does not otherwise occur in the dataset; and

6. performing Unicode normalization according to the NFC standard.

Additionally, the preprocessing script can also be found in the Supplementary Material or at `https://github.com/coastalcph/histnorm`.