Manual and Semi-automatic Normalization of Historical Spelling — Case Studies from Early New High German

Marcel Bollmann and **Stefanie Dipper** and **Julia Krasselt** and **Florian Petran**
Department of Linguistics
Ruhr University Bochum
44780 Bochum, Germany
bollmann,dipper,krasselt,petran@linguistics.rub.de

Abstract

This paper presents work on manual and semi-automatic normalization of historical language data. We first address the guidelines that we use for mapping historical to modern word forms. The guidelines distinguish between *normalization* (preferring forms close to the original) and *modernization* (preferring forms close to modern language). Average inter-annotator agreement is 88.38% on a set of data from Early New High German. We then present *Norma*, a semi-automatic normalization tool. It integrates different modules (lexicon lookup, rewrite rules) for normalizing words in an interactive way. The tool dynamically updates the set of rule entries, given new input. Depending on the text and training settings, normalizing 1,000 tokens results in overall accuracies of 61.78–79.65% (baseline: 24.76–59.53%).

1 Introduction[1]

It is well-known that automatic analysis of historical language data is massively hindered by the fact that such data shows large variance with regard to spelling. Characters and symbols used by the writer of some manuscript reflect impacts as different as dialect influences or spatial constraints. This often leads to inconsistent spellings, even within one text written up by one writer.

In this paper, we present guidelines for manual normalization, which define mappings from historical spellings to modern equivalents. Differences between historical and modern word forms mainly concern graphemic or dialect-specific phonetic/phonological divergencies—which are straightforward to map. The differences further include inflectional and semantic divergencies. In such cases, it is less clear what the goal of normalization should be: We can either stay close to the original and, e.g., keep historical inflection even if it violates modern morpho-syntactic constraints; we call this approach *normalization*. Or else, we can adjust inflection to modern constraints, which we call *modernization*. Of course, (close) normalization is much easier to generate automatically. However, further processing of the data is usually done by taggers and parsers that have been trained on modern data. Hence, data that is maximally similar to modern data would be preferred.

Rather than opting for one of the two forms, we argue for guidelines that serve both camps by providing two levels of normalization.

As already mentioned, historical spelling depends to a large extent on the dialect of the author or printer (or the assumed audience). As a consequence, spelling of historical texts can differ considerably between texts, too. We therefore think that normalization systems should be easily adaptable to specific texts. Our tool *Norma*, presented in this paper, implements such a system, in the form of a semi-automatic normalization tool.

The paper is organized as follows. Sec. 2 addresses related work, Sec. 3 describes the corpora that our studies are based on. Sec. 4 presents the guidelines for manual normalization. In Sec. 5, the tool *Norma* is introduced, followed by an evaluation in Sec. 6. Sec. 7 presents the conclusion.

[1]The research reported here was financed by Deutsche Forschungsgemeinschaft (DFG), Grant DI 1558/4-1.

2 Related Work

There has been considerable work on historical corpora for some time (e.g., the Penn Corpora of Historical English, the Perseus and TITUS corpora, or ARCHER[2]), and increasingly so in the past few years, with the advent of historical corpora for many languages (such as Dutch or Portuguese). Still, guidelines for normalization of spelling variants are often not an issue—e.g., because the corpora are based on editions that standardized spelling to a sufficient extent—or they are not (or not yet) published. This leads to unnecessary duplication of work.

The guidelines developed in the GerManC project (Scheible et al., 2011; GerManC Project, 2012) provide a modern lemma form, using the spelling of the modern dictionary *Duden* or, for obsolete words, the leading forms in *Deutsches Wörterbuch* by Grimm, a historical dictionary. An inflected normalized form is created by attaching suitable modern inflection to the lemma. This form corresponds closely to the modernized form as defined in our guidelines (see Sec. 4.2).

An interactive tool that aids the normalization process is VARD (Baron and Rayson, 2008). It uses a lexicon to detect spelling variants and tries to find modern cognates by a combination of user-defined replacement rules, phonetic matching, and Levenshtein distance. Similarly to the *Norma* tool described in Sec. 5, VARD can be trained by the user confirming or correcting the suggested word forms. However, it is less flexible in that the normalization methods are mostly fixed; e.g., phonetic matching is hard-coded for (Early Modern) English and therefore not suited for German texts, but cannot be turned off or modified. Also, different normalization methods cannot be added.

3 Corpora

To evaluate both applicability of our guidelines and performance of the interactive tool, we created a corpus containing different types of texts: they are written in different dialects, are manuscripts or prints, and from different domains. One part of the texts are fragments of the *Anselm Corpus* (Sec. 3.1), another part comes from the *LAKS Corpus* (Sec. 3.2).

3.1 Anselm Corpus

The *Anselm Corpus* consists of all German manuscripts and prints of the text "Interrogatio Sancti Anselmi de Passione Domini" ('Questions by Saint Anselm about the Lord's Passion'). In the 14th–16th centuries, this text was written down in various German dialects (from Upper, Central, and Low German) and transformed into long and short prose and lyric versions. In total, there are more than 50 German manuscripts and prints, which makes the text an exceptionally broadly-documented resource. The texts are transcribed in the context of an interdisciplinary project.[3] The transcriptions are diplomatic, i.e., they stay maximally close to the original.

For our study, we selected fragments of 1,000 tokens of four manuscripts and two prints from different dialectal regions, and a 4,500-token fragment of a manuscript. The vast majority of the texts are written in Upper German. There are only two Anselm texts in Central German in our collection, and one of them (COL_p) in fact shows many characteristics of Low German. All texts are from the 15th century, i.e., from the period of Early New High German (ENHG). Table 1 provides more information about the texts that we used in our study.[4]

3.2 LAKS Corpus

The LAKS corpus (*Leipziger und Amberger Kanzleisprache*) consists of texts compiled in the chanceries of the medieval German municipalities Leipzig and Amberg. Leipzig is located in the dialectal area of Eastern Central Germany, Amberg belongs to the Eastern Upper German dialectal area. Like the Anselm texts considered in our

[2] Penn Corpora: `http://www.ling.upenn.edu/histcorpora`; Perseus: `http://www.perseus.tufts.edu`; TITUS: `http://titus.uni-frankfurt.de`; ARCHER: `http://www.llc.manchester.ac.uk/research/projects/archer` .

[3] Project partners (and responsible for the transcriptions) are Simone Schultz-Balluff and Klaus-Peter Wegera, Ruhr-University Bochum.

[4] BER_m – Berlin, NUR_m – Nuremberg, SAR_m – Sarnen, WEI_m – Weimar, MEL_m – Melk, AUG_p – Augsburg, COL_p – Cologne, AMB_l – Amberg, LEI_l – Leipzig. These locations indicate the depository in the case of manuscripts, and the place of printing in the case of prints.

Corpus	Text	Size	Type	Dialect	MSTTR±SD	MTLD	MSTTR±SD	MTLD
Anselm	BER$_m$	1,028	ms	ECG	0.701±0.051	73.9	0.701±0.053	78.1
	NUR$_m$	1,003	ms	NUG	0.694±0.053	72.2		
	SAR$_m$	1,022	ms	WUG	0.712±0.047	84.7		
	WEI$_m$	1,003	ms	EUG	0.669±0.035	68.8		
	MEL$_m$	4,536	ms	EUG	0.693±0.044	74.5		
	AUG$_p$	1,022	pr	UG	0.704±0.047	85.4	0.740±0.052	110.9
	COL$_p$	984	pr	ECG	0.767±0.042	131.2		
LAKS	AMB$_l$	1,013	ms	EUG	0.729±0.081	101.7	0.729±0.063	105.7
	LEI$_l$	1,027	ms	ECG	0.733±0.051	99.5		

Table 1: Information on the components of the test corpus; ms = manuscript, pr = print. The dialect abbreviation mainly consists of three letters: 1st letter: E = East, W = West, N = North; 2nd letter: C = Central, U = Upper; 3rd letter: G = German.

study, the LAKS texts were written in the 15th century.

Medieval urban chanceries were concerned with the production of official documents. That included adjudications on disputes between townsmen, settlements on, e.g, inheritance disputes, and further administrative affairs. The terms and arrangements were put down in writing by a municipal clerk: a person with a university degree, excellent writing skills, and a high reputation within the municipality.

The basis of the LAKS corpus are printed editions of the original manuscripts created by Steinführer (2003), Laschinger (1994), and Laschinger (2004). The editions were aimed at an audience of medieval historians; therefore they made minor adjustments, e.g. regarding punctuation, capitalization, and the representation of special graphemes. For our study, we selected two 1,000 token fragments of the Amberg and Leipzig subcorpora, see Table 1.

3.3 Lexical Diversity of the Texts

As measures of lexical diversity, Table 1 reports the Mean Segmental Type-Token Ratio (MSTTR), and Measure of Textual Lexical Diversity (MTLD, McCarthy and Jarvis (2010)) for each individual text as well as for each subcorpus. MSTTR is the average TTR of all segments of 100 tokens in a text or corpus. MTLD is the average number of consecutive tokens it takes to reach a TTR stabilization point of 0.72.

Anselm prints and LAKS texts have higher scores for both metrics than the Anselm manuscripts, which indicates higher lexical diversity. This poses a challenge for the normalization system since higher diversity means that the system cannot generalize the acquired data as well as for the less diverse texts.

4 Normalization Guidelines

For the normalization of historical data, we developed a set of guidelines (Anselm Project, 2012). The main principle is the distinction between *normalization* (Sec. 4.1) and *modernization* (Sec. 4.2). Normalization maps a given historical word form to a close modern cognate, modernization adjusts this form to an inflectionally or semantically appropriate modern equivalent, if necessary.

4.1 Normalization

Normalization as defined here is the transformation of a historical form into its modern equivalent by implementing sound as well as spelling changes. This step presupposes that the word is still part of the modern language's lexicon.

A common sound change from ENHG to modern New High German (NHG) is, e.g., monophthongization of diphthongs. For instance, ENHG /uɐ/ (often spelled <u͡e>) became /uː/ (<u>).

Furthermore, historical word forms often show a high degree of spelling variation due to the lack of a standardized orthography. This variation needs to be mapped onto the modern language's orthography. A common example is the ENHG letter <v>, which is realized as either <w>, <u> or <v> in NHG orthography.

For normalizing proper nouns such as *Judas*, exhaustive lists of modern standard forms are pro-

vided to ensure that they are normalized in the same way. Ex. (1) shows an extract of AUG$_p$, along with its normalization (in the second line).

(1) *Do giench Iudas zv̊ meinem chind*
da ging Judas zu meinem Kind
then went Judas to my child
"Then Judas went to my child"

4.2 Modernization

Not all instances are as easy to normalize as in Ex. (1). Sometimes, the modern cognates created by normalization need to be adjusted inflectionally and semantically, to adhere to present-day syntax and semantics rules. An adjustment of inflection is necessary if inflectional paradigms change over time. Possible changes are the loss of inflectional affixes or changes in inflection class assignment; another example is the change of a word's grammatical gender. We call this type of adjustment 'modernization' to distinguish it from the "pure" (conservative) normalization process described above.

Particularly difficult examples are "false friends": ENHG word forms that look like a modern equivalent but need to be adjusted nevertheless. An example is provided in (i) in Table 2. ENHG *kyndt* 'child(ren)' refers to multiple children in that context, but looks like a singular form from a NHG perspective. The modern equivalent would be *Kinder* (plural) rather than *Kind* (singular). Normalizing tools that operate on individual word forms—as most tools do (but see Jurish (2010))—would probably produce the singular word form. For further processing, the plural word form is to be preferred (otherwise subject–verb agreement is violated). Hence, we decided to retain two forms: the close normalization *Kind* in column NORM (i.e., the modern singular) and an adjusted modernization *Kinder* in column MOD (with the modern plural suffix).

Changes in a word's meaning occur, e.g., due to widening and narrowing, or amelioration and pejoration. To determine the meaning, the context of a given word is crucial (which, again, poses problems for most tools).

Moreover, modernization of a given word form can have an immediate effect on the surrounding word forms, e.g., if the semantically modernized form has a different gender than the historical and normalized forms. Thus, adjacent word forms might need to be adjusted, too. An example is given in (ii) in Table 2. ENHG *vrlaub* has changed its meaning from 'permission' (NHG *Erlaubnis*) to 'vacation' (NHG *Urlaub*). Because *Urlaub* and *Erlaubnis* have different genders, the preceding determiner has to be adjusted, too.

	ENHG	NORM	MOD	Translation
(i)	alle	alle	✓	all
	schult	Schuld	[2]Schulden	debts
	die	die	✓	that
	die	die	✓	the
	fraw	Frau	✓	woman
	und	und	✓	and
	ire	ihre	✓	her
	kyndt	Kind	[2]Kinder	children
	schuldig	schuldig	✓	owing
	sint	sind	✓	are
(ii)	vnd	und	✓	and
	er	er	✓	he
	gab	gab	✓	gave
	Ioseph	Joseph	✓	Joseph
	das	das	[2]die	the
	vrlaub	Urlaub	[1]Erlaubnis	permission
(iii)	vnd	und	✓	and
	zuhant	zehant	[3]sofort	immediately
	nam	nahm	✓	took
	yn	ihn	✓	him
	pilatus	Pilatus	✓	Pilatus

Table 2: Examples for both types of normalization. Columns *NORM* and *MOD* represent (close) normalization and modernization, respectively. If both forms are equivalent, column *MOD* is marked by ✓. Superscribed numbers in column *MOD* indicate semantic (1) and inflection (2) conflicts, and extinct word forms (3).

4.3 Extinct Word Forms

In some cases, no close cognate exists for a historical word form. In that case, we decided to annotate a "virtual" historical word form as the normalized form, along with a suitable NHG translation as the modernized form.

To determine the virtual word form, annotators are asked to consult, in a given order, a range of printed dictionaries[5] to look up the standardized

[5]In our case:

lemma forms. Suitable modern inflectional endings are added to these lemmas.

An example is provided in (iii) in Table 2. The ENHG word form *zuhant* 'immediately' has no NHG equivalent. Hence, it is first normalized by the Lexer ENHG lemma *zehant*. Second, it is translated to the modern equivalent *sofort* (also meaning 'immediately').

4.4 Annotation Results

In our test corpus, 10% of the words were assigned different normalized and modernized word forms in the manual annotation. Among these, 50% are inflection conflicts, 24% semantic conflicts, and 26% concern extinct forms.

For the evaluation of the guidelines, four samples from the Anselm corpus were normalized and modernized manually by two annotators trained on our guidelines. The four samples have a length of 500 tokens each, represent four different dialectal areas, and differ regarding their content. We calculated percent agreement between the two annotators, see Table 3.

	Agreement	
Text	NORM	MOD
BER_m	91.47%	92.06%
NUR_m	85.51%	84.49%
SAR_m	88.02%	89.02%
WEI_m	88.42%	87.82%
Avg.	88.38%	88.38%

Table 3: Inter-annotator agreement on Anselm manuscripts

Average agreement for both tasks is 88.38%. This shows that normalizing is a nontrivial task. However, frequent errors include inflectional adjustments erroneously marked in the NORM rather than the MOD column by one of the annotators.

5 Normalization Tool *Norma*

Norma is a tool for automatic or semi-automatic normalization of historical texts. It is intended to be flexible, as it can be used with any normalization method or a combination of such methods, and can be used both interactively and non-interactively. It supports normalization methods with a trainable set of parameters, which can be dynamically retrained during the normalization process, thereby implementing an incremental learning approach. At the time of writing, only a command-line interface is available.

5.1 Description

Normalization in the *Norma* tool is a modularized process, where each module—or "normalizer"—represents an automatic normalization method. The actual normalizing (and training) methods are implemented individually within each normalizer; *Norma* does not provide this functionality by itself, but rather invokes the respective methods of its normalizer modules.

A normalizer takes a historical word form as input and outputs a suggested modern equivalent along with a confidence score. Confidence scores are numerical values between 0 and 1; a confidence score of 0 is taken to mean that the normalizer failed to find any normalization. In order to normalize a given input word form, multiple normalizers can be combined to form a "chain"; i.e., if the first normalizer fails to find a modern equivalent, the second normalizer in the chain will be called, and so on. An example configuration is presented below. As soon as a normalizer finds an acceptable modern equivalent, this is considered the final normalization, and the chain is stopped. If no modern equivalent is found at all, the original word form is left unchanged.

This method is comparatively simple when compared to VARD (Baron and Rayson, 2008), which chooses the best candidate by calculating an f-score from all normalizers' suggestions. Further extensions to *Norma* are conceivable to allow for more sophisticated combinations of normalizers, but are currently not implemented.

Each normalizer may also utilize a set of parameters and implement a method to dynamically train them. The training method is given both a historical word form and its modern counterpart, which can then be used by the normalizer to adjust its parameters accordingly. This way, normalizers can be adapted to different types of texts, di-

1. Lexer: `http://woerterbuchnetz.de/Lexer`
2. Deutsches Wörterbuch by Jacob and Wilhelm Grimm: `http://woerterbuchnetz.de/DWB`
3. Deutsches Rechtswörterbuch: `http://drw-www.adw.uni-heidelberg.de/drw`

alects, or even languages.

Norma can be used in three different modes: batch, training, and interactive mode. In batch mode, texts are normalized as described above without any user interaction. For training mode, an input file containing both historical and modern word forms must be given, which will then be used to train the normalizers. In interactive mode, the input text is processed step-by-step: for each historical word form, the user is presented with the suggested normalization, which can then be either confirmed or corrected. In either case, the resulting pair of historical and modern word form is passed on to the normalizers' training methods. This represents an incremental learning approach and should ideally improve the results over time, gradually reducing the amount of manual corrections the user has to make.

5.2 Example Configuration

For our own experiments, we used a configuration with two normalizers: a *wordlist substitution* engine; and a *rule-based normalizer*. All input texts were pre-processed to plain alphabetic characters (cf. Bollmann et al. (2011)) and converted to lower case.

Wordlist substitution is one of the simplest approaches to normalization: historical word forms are looked up in a wordlist, where they are directly mapped to one or more modern equivalents. Mappings can trivially be learned from training data; additionally, each mapping is enriched with information on how many times it was learned. This way, whenever a word form is mapped to more than one modern word form, a decision can be made based on which mapping is the most frequent one. Consequently, the confidence score is calculated by dividing the frequency of the chosen mapping by the summarized frequency of all mappings for the given word form.

When a historical word form cannot be found in the wordlist, the *rule-based normalizer* is invoked. The rule-based approach is described in detail in Bollmann et al. (2011). Its main idea is to apply a set of character rewrite rules derived from training data to a historical word form, thereby producing the modern spelling of the word. These rewrite rules operate on one or more characters and also take their immediate context into account. Ex. (2) shows a sample rule.

(2) v → u / # _ n
('v' is replaced by 'u' between the left word boundary ('#') and 'n')

Input word forms are processed from left to right, with one rewrite rule being applied at each position according to a probability score, which also determines the confidence score of the generated word form. One additional restriction is imposed on the final output word form: to prevent the generation of nonsense words, each generated word form is checked against a (modern) *dictionary*. Word forms not found in the dictionary are discarded, so that only words contained in the dictionary can ever be generated by this method.

Learning rewrite rules from training data is done via a modified algorithm for calculating Levenshtein distance, which—instead of simply counting the number of edit operations—keeps track of the exact edit operations required to transform the historical wordform into its modern equivalent.

Note that neither method currently takes token context into account; word forms are only considered in isolation. Due to the sparseness of our data, it is unclear whether including context information can actually improve overall accuracy. However, Jurish (2010) has used token context with promising results, so this is a possible line of future research.

6 Evaluation

To evaluate the performance of *Norma* with its current modules, we manually normalized and modernized six Anselm text fragments and two LAKS fragments of around 1,000 tokens each, and one Anselm text of 4,500 tokens (see Table 1). For the evaluation, we tested different scenarios:

(i) Normalization vs. modernization: Full modernization often needs contextual information, e.g., to adjust inflection to modern usage (see Sec. 4.2). Since our tool currently operates on individual word forms only, we expect considerably higher accuracy with normalized data as compared to modernized data.

(ii) Retraining vs. training from scratch: We investigated whether building upon replacement

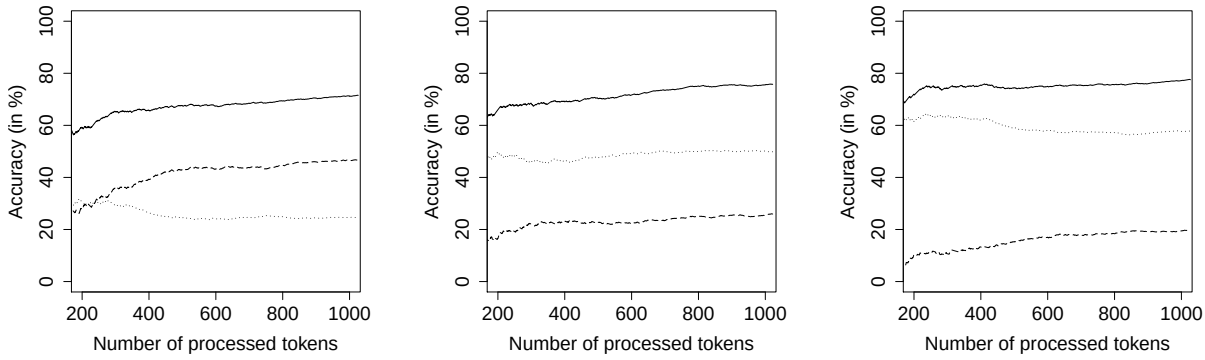

Figure 1: Learning curves from different text types and scenarios: left: BER_m (Anselm manuscript), center: AUG_p (Anselm print), right: LEI_l (LAKS). The solid line (on top) indicates accuracy, the dotted line (in light gray) is the baseline (identity mapping), and the dashed line shows the net learning curve (= accuracy minus the baseline).

rules that were derived from Luther's bible from 1545 (see Bollmann et al. (2011) for details) would facilitate the normalization task.[6] Since the language used by Luther is already quite close to modern German, these rules might not be of much help, since they depend on the idiosyncracies of the evaluated texts.

(iii) Modern dictionary: We also experimented with different dictionaries that the generated word forms would be checked against. In one scenario, the complete vocabulary of a modern Luther bible was used. In the second scenario, the bible wordlist was complemented by a full-form lexicon, consisting of all simplices that can be generated by the German morphology *DMOR* (Schiller, 1996). Finally, as a kind of "upper bound", we added all modern word forms of the test texts to the bible-DMOR lexicon, so that the final lookup would never (or rarely) fail.

Table 4 lists the five overall best and worst results (without using the upper-bound dictionary). The COL_p text is the hardest one, showing lots of characteristics of Low German.

The differences between training from scratch and retraining on bible rules could be used as an indication as to how close the text's language is to Luther's language: if accuracy improves a lot thanks to retraining as opposed to training from scratch, the text must be close to Luther. It turns out that WEI_m and NUR_m profit most from retraining, whereas COL_p and BER_m show small improvement only. This suggests that Luther's language is more similar to Upper than Central German.

For each text, we created *learning curves* that show how much manual input is needed to arrive at an accuracy of *n* percent in these different scenarios. Fig. 1 shows the learning curves for one selected text of each subcorpus (Anselm manuscript, Anselm print, LAKS manuscript). The setting in all three cases was: (i) normal-

Corpus	Norm	Training	Dict	Acc.
LEI_l	norm	retrain	bible	79.65%
LEI_l	norm	retrain	b+d	79.23%
NUR_m	norm	retrain	bible	78.13%
NUR_m	norm	retrain	b+d	77.83%
AUG_p	norm	retrain	bible	77.79%
SAR_m	mod	scratch	bible	64.05%
COL_p	mod	retrain	bible	63.43%
COL_p	mod	retrain	b+d	62.50%
COL_p	mod	scratch	bible	62.09%
COL_p	mod	scratch	b+d	61.78%

Table 4: Overall best and worst results, with different texts and settings; "b+d" refers to the bible dictionary augmented by DMOR forms.

[6] 5.5 million rule instances of about 10,000 types have been learned from Luther, while each of our short text fragments yields only between 8,500–11,000 instances of 1,200–1,500 types.

ization (rather than modernization); (ii) training from scratch; (iii) use of the bible dictionary.

Accuracy (displayed by a solid line) is computed as the ratio of correctly normalized tokens divided by the total number of tokens treated so far (punctuation marks were excluded from the set of tokens). The plots include a simple baseline (displayed by a dotted line): accuracy of a normalizer that does not modify the input word form at all. This is equal to the number of historical word forms that are identical with the corresponding modern word form (and shows how close the text's language is to modern German). Finally, a dashed line indicates the *net* learning curve, which is accuracy minus the baseline. Only results from 200 tokens onward are plotted. [7]

The plot to the left (BER$_m$) shows a rather difficult example of the Anselm manuscripts. The baseline is very low, around 25%. Final overall accuracy (after having normalized 1,000 tokens) is 71.5%. The plot in the center (AUG$_p$) shows the curve of one of the Anselm prints. Here, the baseline is rather high, around 48%. Final overall accuracy is 75.7%. Finally, the plot to the right (LEI$_l$) displays the results of one LAKS manuscript. The baseline is extremely high, around 60%. Final accuracy is 77.5%. In all three cases, overall accuracy as well as net learning curve show a clear steady growth.

Such individual test runs show rather diverse results. In the following, we try to highlight tendencies, by summarizing our main findings.

(i) Normalization vs. modernization As expected, generating normalized rather than modernized word forms is considerably easier. Accuracy improves by 5.0±0.7 (Anselm prints) to 6.1±0.9 (LAKS manuscripts) percentage points for the 1,000-token texts, averaged over all settings. Interestingly, the gap is smaller when more text is available: improvement with the Anselm 4,500-token manuscript is 4.6±0.1.

(ii) Retraining vs. training from scratch All texts profit from the rules derived from the Luther bible. If we compare texts of the same size, the average improvement is smallest with the Anselm prints (accuracy improves by 1.0±0.5 percentage points). Improvements are more pronounced with the Anselm 1,000-token manuscripts (2.2±1.0) and the LAKS manuscripts (2.3±0.9). As can be expected, the differences become less important if the text size is increased: improvement with the Anselm 4,500 text is 0.5±0.2.

(iii) Modern dictionary We observe that the choice of dictionary has less impact on accuracy than the other factors. Average differences are between 0.4±0.3 percentage points (with Anselm manuscripts) and 1.8±1.3 (with LAKS texts). As expected, the "upper-bound" dictionary, which contains all target words, is found most often among the top-10 settings of each subcorpus (in roughly 75% of the cases). However, availability of such a dictionary is certainly not a realistic scenario.

Comparing the original bible dictionary with its DMOR-augmented version, it turns out, surprisingly, that in 69% of the scenarios, the bible dictionary performs better than the augmented version. With the LAKS corpus, however, the augmented version is clearly preferable. This can be attributed to the fact that LAKS, being a corpus of administrative texts, contains many out-of-domain words, which are not covered by the bible dictionary.

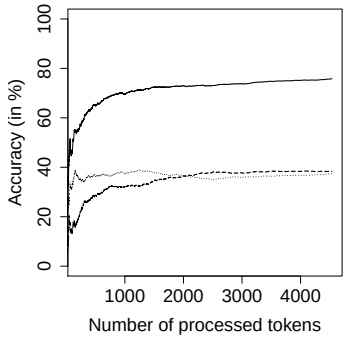

Figure 2: Learning curve for the 4,500 tokens Anselm text (MEL$_m$); lines as in Fig. 1.

The plots in Fig. 1 show that accuracy still improves considerably when more data is added. According to Baron and Rayson (2009), the first 2–3,000 tokens yield a steep increase in performance (for recall). We therefore normalized one entire Anselm text, MEL$_m$, with 4,500 tokens, see Fig. 2. The plot seems to suggest that the "turn-

[7] Below 200, the accuracy fluctuates too much to be meaningfully plotted.

ing point", after which performance increases less rapidly, is already reached after 1,000 tokens.

Correlating accuracy with lexical diversity (Sec. 3.3), it turns out that more diverse texts (Anselm prints and LAKS manuscripts) achieve higher accuracies. However, this can be attributed to the fact that their baselines are also higher in general. In fact, less diverse texts (Anselm manuscripts) show larger increases of accuracy over their baselines (Anselm manuscripts: 35.4±6.5 percentage points; Anselm prints: 31.8±5.7; LAKS manuscripts: 17.9±3.4).

7 Conclusion

In this paper, we presented normalization guidelines, and a semi-automatic normalization tool. We have argued for a two-step approach to normalization: one level ("normalization") which stays formally close to the original, and another level ("modernization") which approximates modern language. Automatic generation of normalized forms is considerably easier, with improvements between 5–6 percentage points in accuracy.

In an ideal setting, both levels of normalization would be generated. The second level would serve further processing like morphological tagging. Mismatches between the first and the second level could provide important hints for inflectional and semantic changes between ENHG and modern German.

The tool *Norma* integrates lexicon lookup and rewrite rules to generate modern word forms that can be corrected in an interactive way. Corrections are used to retrain the methods, improving further normalization suggestions. An evaluation showed this approach to be promising, as accuracy increases considerably with even small amounts of training data. However, accuracy was also found to depend to a great extent on the specific text and the setting.

Possible further research includes a more exhaustive evaluation of different normalization methods and combinations of such methods in particular, for which the *Norma* tool provides an ideal framework. Furthermore, we showed that instead of training normalizers from scratch, it is often preferable to build upon previously learned data, even if it stems from a slightly different domain. How to best combine data from texts of different lengths, types, and/or dialects in order to improve the results on texts for which no special training was available is still an open question.

References

Anselm Project. 2012. Guidelines for Manual Normalization of Historical Word Forms. Manuscript, Ruhr University Bochum.

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*.

Alistair Baron and Paul Rayson. 2009. Automatic standardization of texts containing spelling variation. How much training data do you need? In *Proceedings of the Corpus Linguistics Conference*.

Marcel Bollmann, Florian Petran, and Stefanie Dipper. 2011. Applying Rule-Based Normalization to Different Types of Historical Texts — An Evaluation. In Zygmunt Vetulani, editor, *Proceedings of the 5th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, pages 339–344, Poznan, Poland.

GerManC Project. 2012. GerManC annotation guidelines. Manuscript, University of Manchester.

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

Johannes Laschinger. 1994. *Denkmäler des Amberger Stadtrechts. 1034–1450*, volume 1. Beck, München.

Johannes Laschinger. 2004. *Denkmäler des Amberger Stadtrechts. 1453–1556*, volume 2. Beck, München.

Philip M. McCarthy and Scott Jarvis. 2010. MTLD, voc-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior Research Methods*, 42:381–392.

Silke Scheible, Richard J. Whitt, Martin Durrell, and Paul Bennett. 2011. A gold standard corpus of Early Modern German. In *Proceedings of the Fifth Linguistic Annotation Workshop*, pages 124–128.

Anne Schiller. 1996. Deutsche Flexions- und Kompositionsmorphologie mit PC-KIMMO. In Roland Hausser, editor, *Proceedings of the first Morpholympics*. Tübingen: Niemeyer.

Henning Steinführer. 2003. *Die Leipziger Ratsbücher 1466–1500. Forschung und Edition*, volume 1–2. Leipziger Universitätsverlag, Leipzig.