Normalising Slovene data: historical texts vs. user-generated content

Nikola Ljubešić[*†], **Katja Zupan**[♢*], **Darja Fišer**[‡*], **Tomaž Erjavec**[*♢]
[*] Dept. of Knowledge Technologies, Jožef Stefan Institute
[♢] Jožef Stefan International Postgraduate School
[†] Department of Information and Communication Sciences,
Faculty of Humanities and Social Sciences, University of Zagreb
[‡] Dept. of Translation, Faculty of Arts, University of Ljubljana
nikola.ljubesic@ijs.si, katja.zupan@ijs.si,
darja.fiser@ff.uni-lj.si, tomaz.erjavec@ijs.si

Abstract

The paper presents two manually annotated Slovene language text normalisation datasets, one of historical texts and the other of tweets, and proposes several variants of character-based statistical machine translation to normalise the spelling of their words. The systems differ in whether they perform token-level or segment-level normalisation and whether they make use of additional language resources. The systems are evaluated automatically against the gold standard as well as manually, against a newly developed typology of errors intended to analyse in detail the effect of different types of data and different levels of data standardness. The evaluations show that segment-level normalisation can be useful given a high enough level of token ambiguity, that the same system can be used regardless of the data type, and that background resources will always prove useful.

1 Introduction

Processing non-standard data has been one of the core NLP challenges in the past decade. This was, in the first instance, due to intensive digitisation efforts of textual cultural heritage, which have resulted in greater access to historical language and language variants. However, accessing such non-standard data is not a straightforward process. It may be difficult for a modern reader to understand it, let alone search through it without background knowledge of historical word forms. Additionally, non-standard language also degrades the performance of off-the-shelf NLP tools, which are typically trained on contemporary standard language. These problems have re-surfaced with the emergence of micro text such as short text messages (SMS) and Twitter as well as a boom in informal communication through social networks.

In this paper we present two Slovene language datasets, one of historical texts and the other of tweets, and propose several variants of character-based statistical machine translation to standardise the spelling of their words. Section 2 overviews related work, Section 3 introduces the datasets and background resources, Section 4 details the experimental setup, Section 5 discusses the automatic and manual evaluation of the results, and Section 6 gives the conclusions and directions for further work.

2 Related work

Numerous methods have been proposed on how to process non-standard textual data, mostly by adding a pre-processing step in the form of normalisation into a standard, canonical word form, which brings the non-standard word forms closer to the readers as well as NLP tools.

Our work falls within the framework of character-based statistical machine translation (CSMT), which was first used for transliteration (Matthews, 2007) and then proposed for word normalization tasks as well.

CSMT on automatic word alignments and small training sets has been successful for modernising both historical Icelandic and Swedish (Pettersson et al., 2013). While Sánchez-Martínez et al. (2013) bootstrapped Spanish historical-to-modern lexica from corpora when no word-aligned training data was available with good results, Scherrer and Erjavec (2013) used a large lexicon of modern Slovene to identify the most similar contemporary equivalent for each unknown historical expression, thus improving tagging and lemmatization performance. Pettersson et al. (2014) obtained consistently superior results with CSMT compared to simplistic filtering or Levenshtein distance for four out of five tested languages.

Non-standard words have been receiving attention also in the context of computer-mediated communication (CMC), where erratic punctuation, misspellings, expressions in foreign languages, colloquial and dialectal expressions make it difficult to process (Sproat et al., 2001). Normalisation of CMC was also approached as a translation task (Aw et al., 2006). Li and Liu (2012) combined a single-step CSMT system with a two-stage character/phone translation method to leverage phonetic information. Pennell and Liu (2011) trained a CSMT model for expanding SMS abbreviations in English. Ljubešić et al. (2014) extended the task to all out-of-vocabulary (OOV) tokens by training a model on a manually normalised lexicon of the most salient, Twitter-specific OOV tokens. De Clercq et al. (2013) showed that a cascaded SMT system of a token-based module followed by translation at the character level gives the best word error rate reduction.

All of the above-mentioned CSMT systems perform normalisation at the token level, thus not taking into account contextual information, which could potentially lead to better performance. Successfully experimenting with token-level as well as segment-level systems is the first contribution of this paper, where, by segment-level, we mean stretches of text longer than a single token, e.g., a line or a sentence of the text. The other contributions are a uniform CSMT method obtaining best results on all datasets, regardless of the type of data or their level of standardness, and a significant positive impact of exploiting additional target-language resources.

3 Datasets

This section details the four datasets used in the experiments. They consist of easy and hard cases for normalising words in a historical setting, and in a social media one. We introduce the diachronic dataset, the user-generated one, define our notion of normalisation, and quantify the datasets. Next, the target language models, i.e. datasets used for modelling contemporary standard Slovene, are introduced.

3.1 The historical datasets

A part of the IMP language resources for historical Slovene (Erjavec, 2015a) is the goo300k manually annotated corpus (Erjavec, 2015b), comprising transcriptions of 1,100 pages (about 300,000 tokens) sampled from 88 books and one newspaper, which were published from 1584 to 1899. Each word token in the corpus is annotated for its normalised (modernised) word form(s), their part-of-speech, lemma, and — for archaic words — its gloss, i.e. contemporary synonyms. The corpus has already been used in several word modernisation experiments (Scherrer and Erjavec, 2016; Etxeberria et al., 2016).

The modern-day Slovene alphabet (called the Gaj alphabet, modelled after the Croatian alphabet by Ljudevit Gaj) was introduced into Slovene print in the 1840s; before that, the Bohorič alphabet, modelled on the German one, was used. The introduction of the Gaj alphabet was also closely preceded by a new grammar and subsequent standardisation of the language, therefore the change in the alphabet makes a convenient split between very non-standard and slightly non-standard historical language. As each text in goo300k is marked for its language variant this split is also trivial technically. After removing 3 outlier texts, we extracted from goo300k the following two datasets:

- **Bohorič**: texts written in the Bohorič alphabet published after 1750, as we have only a handful of pages from older texts which are simultaneously much harder to normalise;
- **Gaj**: texts written in the Gaj alphabet, up to 1899, which are the youngest texts in goo300k.

3.2 The social media datasets

The Janes corpus of Slovene CMC (Fišer et al., 2015) contains texts from various internet and social media platforms including Twitter. This subcorpus collects tweets of 8,750 Slovene users who have posted 7.5 million tweets with over 100 million tokens.

While tweets contain a fair amount of very non-standard text with dialectal forms, removed diacritics, phoneticised English etc., many are also completely or mostly standard. We developed a method (Ljubešić et al., 2015) to automatically classify tweets (and other texts) into three levels of technical and linguistic standardness. Technical standardness (T1, quite standard – T3, very non-standard) relates to the use of spaces, punctuation, capitalisation and similar, while linguistic standardness (L1 – L3) takes into account the level of adherence to the written norm and more or less conscious decisions to use non-standard language,

involving spelling, lexis, morphology, and word order. All tweets in the corpus have been labelled with their two standardness scores, while the authors of tweets have been manually classified into corporate ones – such as news agencies, public institutions, companies etc. – and private individuals.

On the basis of these two criteria we prepared the Twitter easy and hard datasets, both containing only private tweets:

- **L1**: 1,000 randomly sampled T1L1 tweets + 1,000 randomly sampled T3L1 tweets
- **L3**: 1,000 randomly sampled T1L3 tweets + 1,000 randomly sampled T3L3 tweets

These tweets were automatically tokenised and normalised, which was then checked and corrected manually by a team of students. The tokenisation and normalisation guidelines mostly followed the ones from the IMP project, but with some modifications regarding the differences of the medium (e.g. emoticons, urls). The annotation was performed in WebAnno (Yimam et al., 2013) where each tweet was annotated by two different annotators and then curated by the team leader (Čibej et al., 2016). For tweets that had been automatically generated by certain applications or had not been written in Slovene, the annotators had the option to mark them as irrelevant for the task.

3.3 Normalisation

What exactly constitutes a "normalised" word is a complex question, and various approaches have been proposed (Eisenstein, 2013). Most, including ours, normalise a word token only orthographically, in the trivial case into the Gaj alphabet, either from the Bohorič alphabet or from non-diacriticised text (c, s, z instead of č, š, ž), which is a common way of entering text on mobile platforms. More generally, archaic or phonetic spellings are also normalised to their standard equivalent. However, we do not substitute extinct, dialectal or slang words with their standard (near)equivalents, but only modify their spelling. This is a similar approach to Bollmann et al. (2012), who distinguish normalisation from modernisation, with the latter also changing the word to its closest modern standard equivalent as regards its morphosyntax and semantics. In cases of orthographic variation of extinct or non-standard words, we normalise them to their most common form in the relevant corpora.

In our work we map spans of original tokens into spans of normalised tokens, with further linguistic annotation assigned to the normalised ones. In the majority of cases, there is 1-1 mapping between the original and the normalised form but the contemporary standard as regards what constitutes an orthographic word also differs in some cases from past practice or that found on social media. Other approaches have typically taken a more restricted approach to normalisation, either always normalising only 1-1 (Han and Baldwin, 2011), or normalising 1-n, but not n-1 cases (Bennett et al., 2010).

To illustrate, we give in Figure 1 two cases, one from the goo300k corpus and the other from the Janes-Tweet subcorpus, both as encoded in the TEI P5 format we use for encoding our corpora. Note that here both are also lemmatised and PoS tagged, but this information is not used in the current experiments.

```
<w lemma="jagoda" ana="#Ncf">jagod</w>
<c> </c>
<choice>
 <orig>
  <w>nar</w>
  <c> </c>
  <w>več</w>
 </orig>
  <reg>
   <w lemma="veliko" ana="#Rgs">največ</w>
  </reg>
</choice>
<c> </c>
<w lemma="bolan" ana="#Agp">bolnih</w>

<w lemma="@chatek" ana="#Xa">@chatek</w>
<c> </c>
<choice>
 <orig>
  <w>Nene</w>
 </orig>
 <reg>
  <w lemma="ne" ana="#Q">ne</w>
  <c> </c>
  <w lemma="ne" ana="#Q">ne</w>
 </reg>
</choice>
<pc lemma="," ana="#Z">,</pc>
<c> </c>
```

Figure 1: Encoding of the normalised corpora. The first goo300k example maps "jagod nar več bolnih" to "jagod največ bolnih", while the second from Janes-Tweet maps "@chatek Nene, " to "@chatek ne ne, ".

3.4 Dataset sizes

Table 1 quantifies the datasets that will be used in the experiments. The first line gives the number of (sampled) texts, where the Bohorič dataset only contains pages from 15 books, while Gaj has pages from almost 70. With L3 and L1 one text is simply one tweet, so the numbers are correspondingly larger. The next line gives the number of original tokens in each dataset; it should be noted that we count cases where *n* original tokens map to one normalised token as one token. Here, by far the largest is the Gaj dataset with almost 250,000 tokens, while the others are of comparable size of about 50,000 tokens. We next give the numbers of tokens that have been normalised (we do not take into account differences in capitalisation), with the next line giving these numbers as percentages of all the tokens. With Bohorič almost half of the tokens needed normalisation, which is of course also due to the differences in the alphabet. With Gaj only about one tenth needed to be normalised, less than in the L3 Twitter dataset, where the number is almost 17%. Finally, L1 is, of course, the most like standard Slovene, with about 3.3% normalisation. Finally, we also give the number of split or joined words as regards normalisation. These cases pose special technical as well as methodological problems in the process of normalisation, even though the numbers are rather low, with all being less than 1%, while their distribution follows the percentages of normalised tokens.

	Bohorič	Gaj	L3	L1
Texts	15	69	1,983	1,957
Tokens	75,210	249,146	54,694	47,950
Norm.	36,493	29,012	9,203	1,572
	48.52%	11.64%	16.83%	3.28%
Multi.	641	1,093	276	131
	0.85%	0.44%	0.50%	0.27%

Table 1: Sizes of the four datasets.

3.5 Splitting the datasets

For our experiments we split each of the four datasets into training, development, and test parts following a 80:10:10 ratio. Sampling was performed by shuffling on segment, i.e. sentence level.

Having development data was necessary as SMT systems without tuning, i.e. with default parameter values, regularly underperform in comparison to tuned systems.

In the interests of replicability of experiments the pre-processed data with our splits is published via the CLARIN.SI language resource repository, c.f. Ljubešić et al. (2016).

3.6 Target language datasets

While additional parallel data for SMT is expensive and therefore hard to acquire, including bigger target language models, which regularly improves translation quality, is a rather simple task as for most target languages there are monolingual resources available. In our experiments we used two corpora of our target language, standard contemporary Slovene, of different quality, size, and costs of construction.

Web corpora are cheap to acquire and can be quite large, and we used **slWaC** (Ljubešić and Erjavec, 2011), a one billion token corpus crawled from the *.si* top level domain, using language identification to filter out non-Slovene texts.

However, Web corpora are noisy and also contain non-standard language, which e.g. is not diacriticised, potentially leading to low-quality models of standard Slovene. This is the reason we also use **Kres** (Logar Berginc et al., 2012), a 100 million word reference and balanced corpus of contemporary Slovene, which contains, for the most part, proof-read texts.

4 Experimental setup

Our experiments have been carried out with the tools of the standard SMT pipeline: MGIZA[1], a multi-threaded version of GIZA++ (Och and Ney, 2003) for alignment, Moses[2] (Koehn et al., 2007) for phrase extraction and decoding, and KENLM[3] (Heafield, 2011) for language modelling. In particular, we have explored character-based SMT, where a word or a segment of the text is split into individual characters, borders between tokens being encoded with underscores, and the resulting string is then translated.

As will be discussed below, we use two granularities of translation, one of tokens and the other of segments. While segments can, in general, be any contiguous stretch of text, in our experiments segments are sentences.

[1] https://github.com/moses-smt/mgiza
[2] http://www.statmt.org/moses/
[3] https://kheafield.com/code/kenlm/

4.1 Research questions

In this paper we are interested in answering our two main research questions:

1. Is there one single CSMT setting that performs best on text normalisation regardless whether we normalise historical texts or user-generated content?

2. Can we outperform the traditional token-by-token normalisation by translating whole segments at a time, therefore taking into account the context in which a token occurred?

We answer these questions by running the following experiments on each of the four datasets:

- experiment1: comparing token-level and segment-level translators when using language models (LMs) based on training data only;

- experiment2: comparing the token-level and segment-level approaches when including additional LMs.

For token-level systems we use order-7 language models while for segment-level systems we opt for order-10 language models. Our early experiments have shown that these orders yield best results in each of the approaches.

We additionally look into the impact of reordering, traditional part of SMT, and time and memory requirements for each of the approaches.

4.2 Evaluation

We evaluate all our experiments on the level of segments. This means that in case of the token-level translator we translate token by token and then combine these translations into segments before evaluating. During all the experiments we evaluated the segment pairs with two metrics: character-level Levenshtein distance normalised by the length of the reference data and the token-level BLEU metric (Papineni et al., 2002). In the remainder of the paper we report the Levenshtein metric only as it was shown for these two metrics to correlate in all experiments with a Pearson's correlation coefficient greater than 0.99.

We perform statistical significance testing on the Levenshtein evaluation metric by using the approximate randomisation test (Yeh, 2000) with 1000 iterations.

4.3 Baselines and ceilings

In our experiments we use two different baselines: the leave-as-is baseline (LAI), which does not transform the input in any way, and the most-frequent-translation baseline (MFT), which exchanges each token with the token most frequently normalised to in the training data. In the MFT baseline ties are resolved randomly.

We use two ceilings as well, both based on the MFT baseline. These ceilings are informative as to what extent word form transformations are ambiguous, i.e. for what amount of error the only solution is disambiguation in context. The first ceiling, MFT is actually a MFT baseline both trained and tested on the test data. The second ceiling, MFTr is the MFT baseline trained both on training and testing data, while tested on testing data only. We consider the second ceiling to be more realistic as it learns on more than testing data, therefore having a lower probability of measuring rare token transformations as the most frequent ones.

5 Results

5.1 Automatic evaluation

5.1.1 First experiment

In the first set of experiments we train and tune token-level and segment-level translators for each of the four datasets. Additionally, we train translators that use reordering models and those that do not use reordering. Here we report the results for the translators that do not use reordering models as the difference between the systems using and not using reordering has no statistical significance.[4]

Table 2 gives the two baselines and two ceilings along with the results of our eight initial systems.

The LAI baseline draws a clear picture about the level of intervention necessary in each of the texts. While in Bohorič 18% of characters have to be transformed, in the L1 dataset less than 1% needs intervention.

Applying the MFT baseline on hard datasets (Bohorič and L3) resolves more than half of the problems. The lowest error reduction with MFT on the L1 dataset is 17.33%.

The two ceilings show that the level of ambiguity on the token transformation level is actually very

[4] On any of the eight pairs of systems (four datasets, each token- and segment-level), the lowest p-value obtained was 0.076, the second and third being 0.138 and 0.261, in roughly half of the cases reordering was performing better, regardless of the type of translation (token- or segment-level).

	baselines		ceilings		first experiment		
	LAI	MFT	MFT	MFTr	token	segm	Δ
Bohorič	17.63	6.46	0.34	0.44	**1.55**	1.92	-23.9
Gaj	3.13	1.43	0.23	0.29	**1.01**	1.15	-13.9
L3	5.15	2.44	0.37	0.54	2.19	2.12	3.20
L1	0.75	0.62	0.05	0.07	0.41	0.43	-4.88

Table 2: Results of the first set of experiments (no additional LMs) as percentages of character errors. Δ is error reduction (in %) by the segment-level system.

low. While on the L1 dataset there is almost no ambiguity (if we saw enough token transformations, only 0.07 percent of characters would not be normalised correctly), in case of Bohorič and L3 every 200th character would be wrongly normalised.

The results of the first experiments show a very similar performance regardless of whether token-level or segment-level translators were used, with a small but consistent better performance of the token-level systems.

The results on the datasets where a statistically significantly better result was obtained are given in bold. Interestingly, on historical datasets the token-level systems perform significantly better than segment-level systems.

The only dataset in which the segment-level system performs better, although not statistically significant, is the L3 dataset, on which the ceilings are also most distant from a perfect normalisation, i.e. the gain to be obtained by taking into account a token's context is the highest.

5.1.2 Second experiment

We continue our experiments by including additional language models into the translators. The idea behind this second experiment is twofold:

- there is not much training data on which the initial language models are based, and adding easy-to-obtain standard data in the form of additional language models is easy in the case of Slovene as is for most languages;
- segment-level translators need much more target-language data than the token-level ones; our assumption is that the segment-level systems on the datasets where more token-level ambiguity is present (like Bohorič and L3) could win over the token-level systems once they obtain enough context evidence.

The additional language models are built from the Kres and the slWaC corpora, again of order 7 in case of the token-level approach and of order 10 in case of the segment-level approach. We combine language models by adding more entries in the moses.ini file and letting MERT weight each language model on our development data.

We experiment by adding each language model separately to the setting using the existing training data language model, and by using all three language models simultaneously. The results of this set of experiment are given in Table 3.

The results confirmed our assumptions: on Bohorič and L3, where token-level ambiguity is higher, segment-level outperform token-level approaches, while on the Gaj and the L1 datasets the token-level still outperforms the segment-level approach.

On the Bohorič dataset in all three LM settings the segment-level approach outperforms the token-level one, with error reduction spanning from 6% to 12%, in which case the difference between the token- and the segment-level approach is statistically significant. Similarily, on the L3 dataset, once the LM based on web data is added, the segment-level approach obtains better results with error reduction of 7% and 10%, the latter being statistically significant. On the two remaining datasets the token-level approach always performs better, but nowhere with a statistically significant difference.[5]

When comparing best-performing systems using training data only and using additional LMs, regardless of the setting (token- vs. segment-level), the error reduction on the Bohorič dataset reaches 14%, on the Gaj dataset 10%, on the L3 dataset 22% and on the L1 dataset 17%, proving that, regardless of the approach, significant and easy-to-obtain improvements can be achieved by expanding the set

[5]Similar trends were observed in (Scherrer and Ljubešić, 2016) on normalising Swiss German where the MFTr ceiling, calculated as token accuracy, is 93% with an error reduction when moving from token-level to segment-level normalisation of 20%. In our datasets the MFTr ceiling, when calculated as token accuracy, is 94.46% for Bohorič, 96.50% for Gaj, 93.88% for L3 and 98.37% for L1.

	Bohorič			Gaj			L3			L1		
LM	token	segm	Δ	token	segm	Δ	token	segm	Δ	token	segm	Δ
train	1.55	1.92	-23.4	1.01	1.15	-13.4	2.19	2.12	3.0	0.41	0.43	-4.0
+kres	1.50	1.40	7.2	0.90	0.94	-4.2	1.81	1.82	-0.1	0.38	0.43	-12.0
+slwac	1.48	1.39	6.4	0.91	1.04	-13.3	1.76	**1.58**	10.2	0.37	0.38	-3.4
+both	1.51	**1.33**	11.8	0.91	0.93	-3.1	1.77	1.65	6.6	0.34	0.38	-10.8

Table 3: Results of the second set of experiments (additional LMs). Δ is error reduction (in %) by the segment-level system.

of language models used.

During the second set of experiments we also measured the time and space requirements of decoding with the token-level and segment-level decoders. A reasonable assumption is that both space and time requirements of the segment-level decoder will be orders of magnitude higher as both its language models as well as its search space are much bigger. The time necessary to translate each of the test sets was roughly 3 times longer in case of the segment translator. Regarding the memory requirements, the difference became quite drastic with 25 times more memory consumption of the segment-level translator when all three language models (train+kres+slwac) were used.

Regarding our two main research questions, the answers obtained through these experiments are the following:

1. regardless of the type of text to be normalised, using the baseline Moses setting with removed reordering (no lexical reordering model and distortion set to zero) and additional language models yields best results

2. if the level of token ambiguity is high, segment-level translation can give significant improvements in translation quality, but with a heavy hit on time and memory requirements

5.2 Manual evaluation

To obtain a better insight into the errors, we made a manual evaluation and comparison between the best token-based and the best segment-based normaliser.

A sample of random 100 word forms incorrectly normalised by at least one of the two normalisers was selected from each of the four datasets.[6] The errors in these 399 instances were manually categorised into 8 types, with the error types chosen with respect to their potential for introducing improvements in the method of normalisation:

- **XF**: corruption of foreign language words (mostly German, Latin or English; e.g. 'interne' instead of 'intern'); here (word or span-level) language identification would be helpful in preventing such wrong normalisations to contemporary standard Slovene;

- **TR**: transliteration error, either from Bohorič or by failing to rediacriticise for L1 and L3 (e.g. 'tisina' instead of 'tišina' (silence)); a special module for rediacriticisation, such as Ljubešić et al. (2016), could reduce these errors;

- **WB**: a word boundary error (e.g. 'naj lepši' instead of 'najlepši'); these are interesting as they are by definition outside the scope of the token-based normaliser;

- **END**: an error in the inflectional ending (e.g. the normaliser failing to change the archaic adjectival suffix "-iga" into the contemporary "-ega", i.e. "lepiga" instead of "lepega"); such errors could be taken care of by introducing suitable morphological processing into the language model;

- **LEX-D**: an error where the wrong word form was predicted, but this word form does in fact exist, however, it belongs to a different part of speech from the correct one (e.g. preposition 'k' (to) instead of conjunction 'ko' (when)); these errors could be alleviated by having a POS tagger determine the expected POS of the target word;

- **LEX-S**: same error as LEX-D, except that the predicted word has same part of speech as the correct one (e.g. preposition 'o' (about) instead of preposition 'ob' (by)); these are

[6] To be exact, only 99 word form errors were taken from L1, as there were only that many errors in this dataset.

among the more intractable errors, and could be resolved only by having access to some sort of word sense disambiguation;

- **VAL**: a (lexical) validity error, where the predicted word does not exist, although it does follow the spelling conventions of contemporary standard Slovene (e.g. 'izdatelj' instead of 'izdajatelj' (publisher)); such errors could be prevented by having a representative lexicon against which to filter hypotheses;

- **OTH**: multiple errors, or errors that could not be categorized into any of the categories listed above (e.g. 'po semi' instead of 'pozimi' (adverb meaning during the winter); 'Avstri' instead of 'Avstrija' (Austria)); these would probably not be corrected even if all of the above-mentioned extra modules were in place.

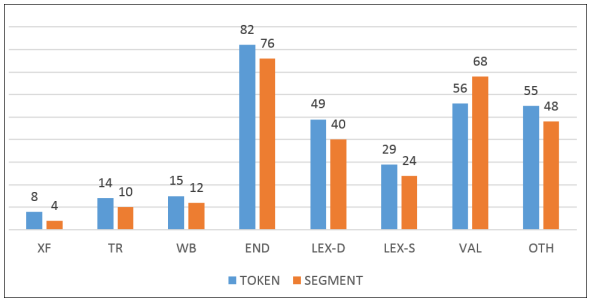

Figure 2: Token vs. segment normalisation on combined datasets.

Figure 2 shows the comparison between the token-based and segment-based systems with all datasets combined. It shows that the segment-based approach outperforms the token-based one in all error types but one, i.e. in the category of validity errors. Only in this case, taking context into account hurts rather than helps: by staying limited to tokens, i.e. word forms, more valid guesses — albeit not necessarily entirely correct — are produced. In total, the token-based system made 308 errors in the analysed sample, while the segment-based one committed 282 errors.

The analysis in Figure 3 shows the distribution of errors made by the segment-based system as the better performing normaliser.

Errors related to foreign words, transliteration, word boundaries, lexical homographs (different POS) and non-categorized errors are more frequent

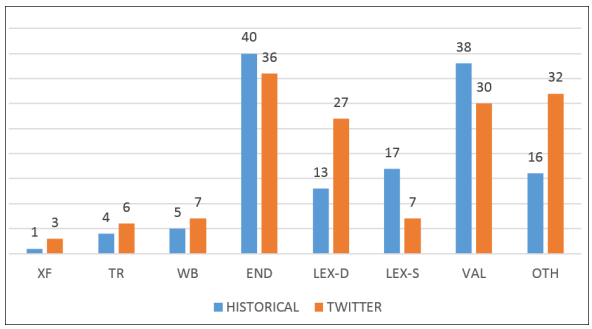

Figure 3: Normalisation on historical and Twitter datasets for the segment-based system.

when normalising tweets, while inflectional endings, lexical homographs (same POS) and validity errors are more typical when normalising the historical datasets. This is to be expected, as many of these errors stem from diachronic changes from historical to modern Slovene.

In both datasets the prevailing type of error is inflectional endings (26.8% of total errors), with the second most common, again for both datasets, being lexical validity, where the normalisers proposed a non-existing word. If we exclude Other errors, the third most common error type is incorrect but existing words with the wrong POS. Interestingly, there are very few errors related to foreign words, transliteration and word boundaries, so from an accuracy point of view, it is not worth investing resources into fixing them.

6 Conclusions

The paper presented experiments in normalising words in historical and user-generated Slovene texts, additionally investigating the differences between cases of easy and hard normalisation. We used CSMT for the task, where we investigated the differences between token-level and segment-level normalisation as well as (not) using additional background resources for better probability estimates regarding the target language.

The experiments show that if token-level ambiguity, measured by training the most-frequent-translation system on both training and testing data and calculating normalised character-level Levenshtein distance, is above 0.04, training a segment-level system could prove to be useful. This, naturally, does not depend on the level of token ambiguity only, but on the amount of parallel and target-side data as well. By applying segment-level approaches on the two datasets with higher token-

level ambiguity, we achieved error reduction of more than 10%. Adding more language models should always be considered as this is not a costly task, and error reductions on our datasets reached between 10% and 22%. Additionally we have also shown that there is no need to use different systems for historical and modern non-standard texts, as the best performing one caters for both datasets.

We performed a manual error classification of the two best performing systems on all four datasets, which showed that about a quarter of all errors are due to poorly normalised inflectional endings, followed by normalising to non-existent words and then by incorrect but existing words with the wrong POS.

Taking into account the most frequent errors of our current systems, there are two main directions how we can improve our result.

The first direction should focus on enriching surface contextual information, either by including larger language models, language models of higher order, language models of higher-order events like tokens, or language models with better abstraction capabilities like neural language models.

The second direction should focus on a higher linguistic abstraction like morphosyntax. Having enough data to train a reasonable part-of-speech tagger over source data could provide us with reasonable morphosyntactic annotation that could be used in the translation process via factored machine translation. Including factors only on the target side, for which very good morphosyntactic annotation can be obtained, should also be investigated.

Acknowledgments

The research leading to these results has received funding from the Slovenian Research Agency within the national basic research project "Resources, Tools and Methods for the Research of Nonstandard Internet Slovene" (J6-6842, 2014-2017), the Young Researcher programme (no. 37487) and the Swiss National Science Foundation grant no. IZ74Z0_160501 (ReLDI).

References

AiTi Aw, Min Zhang, Juan Xiao, and Jian Su. 2006. A phrase-based statistical model for sms text normalization. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 33–40. Association for Computational Linguistics.

Paul Bennett, Martin Durrell, Silke Scheible, and Richard J. Whitt. 2010. Annotating a historical corpus of German: A case study. In *Proceedings of the LREC 2010 Workshop on Language Resource and Language Technology: Standards - state of the art, emerging needs, and future developments*, pages 64–68, Paris. ELRA.

Marcel Bollmann, Julia Krasselt, and Florian Petran. 2012. Manual and semi-automatic normalization of historical spelling – Case studies from Early New High German. In *In Proceedings of KONVENS 2012 (LThist 2012 workshop*, pages 342–350.

Jaka Čibej, Darja Fišer, Tomaž Erjavec, and Špela Arhar Holdt. 2016. Razvoj učne množice za izboljšano označevanje spletnih besedil (The development of a training set for better annotation of internet texts). In *Conference on Language Technologies and Digital Humanities*, Ljubljana, September.

Orphée De Clercq, Bart Desmet, Sarah Schulz, Els Lefever, and Véronique Hoste. 2013. Normalization of dutch user-generated content. In *9th International conference on Recent Advances in Natural Language Processing (RANLP 2013)*, pages 179–188. INCOMA.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *In Proc. of NAACL.*

Tomaž Erjavec. 2015a. The IMP historical Slovene language resources. *Language Resources and Evaluation*, pages pp. 1–23.

Tomaž Erjavec. 2015b. *Reference corpus of historical Slovene goo300k 1.2*. Slovenian language resource repository CLARIN.SI, http://hdl.handle.net/11356/1025.

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)*, Portorož, Slovenia, may. European Language Resources Association (ELRA).

Darja Fišer, Nikola Ljubešić, and Tomaž Erjavec. 2015. The Janes corpus of Slovene user generated content: construction and annotation. In *International Research Days: Social Media and CMC Corpora for the eHumanities*, Rennes, October.

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 - Volume 1*, HLT '11, pages 368–378, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *In Proc. of the Sixth Workshop on Statistical Machine Translation*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the ACL 2007 demonstration session*, pages 177–80, Prague, Czech Republic.

Chen Li and Yang Liu. 2012. Normalization of text messages using character-and phone-based machine translation approaches. In *Proceedings of InterSpeech*, Portland, Oregon, USA.

Nikola Ljubešić, Tomaž Erjavec, and Darja Fišer. 2014. Standardizing tweets with character-level machine translation. *Computational Linguistics and Intelligent Text Processing*, pages 164–175.

Nikola Ljubešić, Darja Fišer, Tomaž Erjavec, Jaka Čibej, Dafne Marko, Senja Pollak, and Iza Škrjanec. 2015. Predicting the Level of Text Standardness in User-generated Content. In *Proceedings of Recent Advances in Natural Language Processing*.

Nikola Ljubešić, Katja Zupan, Darja Fišer, and Tomaž Erjavec. 2016. *Dataset of normalised Slovene text KonvNormSl 1.0*. Slovenian language resource repository CLARIN.SI. `http://hdl.handle.net/11356/1068`.

Nikola Ljubešić and Tomaž Erjavec. 2011. hrWaC and slWac: Compiling Web Corpora for Croatian and Slovene. In Ivan Habernal and Václav Matousek, editors, *TSD*, volume 6836 of *Lecture Notes in Computer Science*, pages 395–402. Springer.

Nikola Ljubešić, Tomaž Erjavec, and Darja Fišer. 2016. Corpus-based diacritic restoration for south slavic languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), may.

Nataša Logar Berginc, Miha Grčar, Marko Brakus, Tomaž Erjavec, Špela Arhar Holdt, and Simon Krek. 2012. *Korpusi slovenskega jezika Gigafida, KRES, ccGigafida in ccKRES: gradnja, vsebina, uporaba [The Gigafida, KRES, ccGigafida and ccKRES corpora of Slovene language: compilation, content, use]*. Zbirka Sporazumevanje. Trojina, zavod za uporabno slovenistiko: Fakulteta za družbene vede, Ljubljana, Slovenia.

David Matthews. 2007. Machine transliteration of proper names. *Master's Thesis, University of Edinburgh, Edinburgh, United Kingdom*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Deana Pennell and Yang Liu. 2011. A character-level machine translation approach for normalization of sms abbreviations. In *IJCNLP*, pages 974–982.

Eva Pettersson, Beáta Megyesi, and Jörg Tiedemann. 2013. An smt approach to automatic annotation of historical text. In *Proceedings of the Workshop on Computational Historical Linguistics at NODALIDA 2013, NEALT Proceedings Series*, volume 18, pages 54–69.

Eva Pettersson, Beáta Megyesi, and Joakim Nivre. 2014. A multilingual evaluation of three spelling normalisation methods for historical text. *Proceedings of LaTeCH*, pages 32–41.

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. *arXiv preprint arXiv:1306.3692*.

Yves Scherrer and Tomaž Erjavec. 2013. Modernizing historical slovene words with character-based smt. In *BSNLP 2013-4th Biennial Workshop on Balto-Slavic Natural Language Processing*.

Yves Scherrer and Tomaž Erjavec. 2016. Modernising historical slovene words. *Natural Language Engineering*, FirstView:1–25, 5.

Yves Scherrer and Nikola Ljubešić. 2016. Automatic normalisation of the swiss german archimob corpus using character-level machine translation. In *Proceedings of KONVENS 2016*.

Richard Sproat, Alan W Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. 2001. Normalization of non-standard words. *Computer Speech & Language*, 15(3):287–333.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 2*, COLING '00, pages 947–953, Stroudsburg, PA, USA. Association for Computational Linguistics.

Seid Muhie Yimam, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann. 2013. Webanno: A flexible,web-based and visually supported system for distributed annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (System Demonstrations) (ACL 2013)*, pages 1–6, Stroudsburg, PA, USA, August. Association for Computational Linguistics.