Modernising historical Slovene words

Scherrer, Yves; Erjavec, Tomaž

How to cite

SCHERRER, Yves, ERJAVEC, Tomaž. Modernising historical Slovene words. In: Natural language engineering, 2016, vol. 22, n° 6, p. 881–905. doi: 10.1017/S1351324915000236

This publication URL: https://archive-ouverte.unige.ch//unige:82305
Publication DOI: 10.1017/S1351324915000236

Modernising historical Slovene words

Yves Scherrer[1], Tomaž Erjavec[2]
[1] *LATL-CUI, Université de Genève*
7 route de Drize, 1227 Carouge, Switzerland
yves.scherrer@unige.ch
[2] *Dept. of Knowledge Technologies, Jožef Stefan Institute*
Jamova cesta 39, 1000 Ljubljana, Slovenia
tomaz.erjavec@ijs.si

(*Received xxxx; revised xxx*)

Abstract

We propose a language-independent word normalisation method and exemplify it on modernising historical Slovene words. Our method relies on character-level statistical machine translation (CSMT) and uses only shallow knowledge. We present relevant data on historical Slovene, consisting of two (partially) manually annotated corpora and the lexicons derived from these corpora, containing historical word–modern word pairs. The two lexicons are disjoint, with one serving as the training set containing 40,000 entries, and the other as a test set with 20,000 entries. The data spans the years 1750–1900, and the lexicons are split into 50-year slices, with all the experiments carried out separately on the three time periods. We perform two sets of experiments. In the first one – a supervised setting – we build a CSMT system using the lexicon of word pairs as training data. In the second one – an unsupervised setting – we simulate a scenario in which word pairs are not available. We propose a two-step method where we first extract a noisy list of word pairs by matching historical words with cognate modern words, and then train a CSMT system on these pairs. In both sets of experiments we also optionally make use of a lexicon of modern words to filter the modernisation hypotheses. While we show that both methods produce significantly better results than the baselines, their accuracy and which method works best strongly correlates with the age of the texts, meaning that the choice of the best method will depend on the properties of the historical language which is to be modernised. As an extrinsic evaluation we also compare the quality of part-of-speech tagging and lemmatisation directly on historical text and on its modernised words. We show that, depending on the age of the text, annotation on modernised words also produces significantly better results than annotation on the original text.

1 Introduction

Digital libraries containing historical publications are becoming increasingly common, mostly due to national and European cultural heritage projects and to the Google Books initiative (Michel, Shen, Aiden, Veres, Gray, The Google Books Team, Pickett, Hoiberg, Clancy, Norvig, Orwant, Pinker, Nowak, and Lieberman Aiden 2011). While some libraries offer only facsimiles, most add to these either OCRed transcriptions and, for more important works, hand-corrected transcriptions. Making such digital historical texts accessible in the same way as modern texts raises several problems. First, full-text search is problematic, as users of digital libraries will not be aware of all the ways in which a word was written in the past, leading to low recall on queries to the search engine. Second, reading and comprehension will be impaired with older texts, especially in cases where spelling conventions have changed. And, third, it is difficult to automatically annotate the text with linguistic information, in particular with part-of-speech (PoS) tags and lemmas, as there are typically no computational models or manually annotated corpora which could enable such processing. These problems affect not only users of digital libraries, but also linguists using historical corpora in their research: a corpus that has no normalisation over its words, is not (or is very badly) lemmatised and tagged, will not be very useful as a basis for diachronic language study.

A common way around these difficulties is to first modernise the individual words, i.e., to convert them to the modern norm (Piotrowski 2012): this significantly improves text search and comprehension as well as further text processing by allowing PoS tagging, lemmatisation and parsing models trained on modern language to be used on historical texts. While not all problems are solved with this approach (e.g., changes of syntax and word-formation processes are not covered), this is a good first step in processing historical language.

In this paper we propose and analyse methods to modernise words of different historical stages of the Slovene language, spanning from 1750 to 1900. We perform the experiments on an extensive, real-world dataset, which is also made freely available for further experiments. The main approach we investigate is character-level statistical machine translation (CSMT), using the Moses toolkit (Koehn, Hoang, Birch, Callison-Burch, Federico, Bertoldi, Cowan, Shen, Moran, Zens, Dyer, Bojar, Constantin, and Herbst 2007). We propose a method to perform supervised word modernisation and also investigate unsupervised settings.

The rest of this paper is structured as follows: Section 2 presents related work in historical word modernisation and in CSMT. Section 3 details the dataset used, in particular a training and testing corpus, a derived lexicon of historical Slovene, and a lexicon of modern Slovene. Section 4 presents the experiments in supervised and unsupervised settings, where the former use a lexicon of modernisations of historical words and the latter only texts (words) in the historical language and a corpus (lexicon) of the modern language. Section 5 discusses the impact of word modernisation on part-of-speech tagging and lemmatisation, whereas Section 6 gives some conclusions and directions for further research.

2 Related work

2.1 Automatic modernisation of historical language

In the last few years, several methods have been proposed to modernise – or normalise – historical words. Most methods use some combination of lexicon lookup, string edit distance computations and explicit transcription rules which are either hand-crafted or automatically induced from training data. More recently, methods based on the statistical machine translation paradigm have emerged.

One of the first systems was VARD 2 (Rayson, Archer, Baron, and Smith 2007; Baron and Rayson 2008), which views text normalisation as a particular case of spellchecking. The tool, which is freely available and still maintained, is a complete environment for modernising Early Modern English texts and integrates an editor, a rule induction mechanism operating on already corrected texts, and a combination of several mechanisms to guess the modern word from a historical one: using a lexicon of known variants, using letter replacement rules, phonetic matching and edit distance. A similar system has been used for German (Scheible, Whitt, Durrell, and Bennett 2011).

The traditional approach to modernising historical words uses transcription rules (e.g., zh → č), which are applied to historical words, with the results filtered against a reference lexicon of the modern language. In our previous work in developing a "transcription" module for historical Slovene (Erjavec 2011) we first used a fixed lexicon of already known historical word–modern word pairs together with a lexicon of modern language extracted from a large automatically annotated corpus. For unknown words we used the Vaam (Variant approximate matching) library (Reffle 2011), which models the transcription rules as (extended) finite-state automata and, additionally, is also able to use Levenshtein distance for correcting OCR errors. Given an input historical word, the output of Vaam is the set of all the words in the modern lexicon (together with the rules that generated them) that can be produced using the given rule set. Vaam does not rank the hypotheses, but a ranking can be induced, e.g., on the number of rules that were applied, or on frequencies of the proposed words in the modern lexicon.

Jurish (2010) argues that pronunciation is generally more stable over time than spelling. In consequence, he compares word forms on the basis of their pronunciation, using and adapting a phonetisation module from an existing German text-to-speech system. However, pronunciation-based normalisation is not sufficient, and the required resources may not be available for languages like Slovene.

Bollmann (2012) presents a method to automatically extract context-sensitive rewrite rules from a parallel corpus. The historical (Early New High German) and modern (New High German) varieties are word-aligned, and for each pair the Levenshtein alignments are computed. From these alignments, normalisation rules are extracted and ranked according to their frequencies of occurrence in the corpus. The rules may operate on sequences of characters. These rules are used to normalise historical word forms, either on their own or in conjunction with a filtering step based on a reference lexicon of the modern language. A similar approach is presented in Kestemont, Daelemans, and De Pauw (2010), where Middle Dutch word forms from the 12th century are converted to modern lemmas using memory-based learning.

Pettersson, Megyesi, and Nivre (2013) do not use explicit transcription rules, but rather search for each historical word its most similar modern counterpart in a lexicon; similarity is measured with Levenshtein distance. This method does not require any parallel data (i.e., manual normalisations) and can thus be completely unsupervised. They also extend their method by including some supervision in the form of historical–modern word pairs in order to tune distance thresholds and to weight the edit operations. The various extensions are shown to improve modernisation accuracy for Swedish data against the basic unsupervised model.

Normalisation of historical words can also be viewed as a special case of matching cognate words between closely related languages. Cognate word matching has been shown to facilitate the extraction of translation lexicons from comparable corpora (Melamed 1995; Koehn and Knight 2002; Kondrak, Marcu, and Knight 2003; Fišer and Ljubešić 2011; Fišer and Sagot 2015). In this area, a large number of similarity measures have been developed (Tiedemann 1999; Kondrak and Dorr 2004; Kondrak and Sherif 2006), and cognate generation models based on such similarity measures, stochastic transducers or HMMs have been introduced, e.g., Mann and Yarowsky (2001) for closely related languages or Scherrer (2007) for dialects.

2.2 Character-level statistical machine translation

More recently, character-level statistical machine translation (CSMT) has been proposed as an alternative approach to translating words between closely related languages (Vilar, Peter, and Ney 2007; Tiedemann 2009). Character-level SMT is different from standard (i.e., word-level) SMT in that, instead of aligning words occurring in sentence pairs, one aligns characters occurring in word pairs. The resulting translation models contain phrases which consist of character sequences instead of word sequences, and language models are trained on character n-grams instead of word n-grams.

CSMT requires less training data than word-level SMT but is limited to applications where regular changes occur at the character level. It has been successfully used for translation between closely related languages (Vilar et al. 2007; Tiedemann 2009), transliteration (Tiedemann and Nabende 2009), lexicon induction (Scherrer and Sagot 2014), cognate generation (Beinborn, Zesch, and Gurevych 2013), standardisation of user-generated content (De Clercq, Desmet, Schulz, Lefever, and Hoste 2013; Ljubešić, Erjavec, and Fišer 2014) and finally normalisation of historical words (Sánchez-Martínez, Martínez-Sempere, Ivars-Ribes, and Carrasco 2013; Pettersson, Megyesi, and Tiedemann 2013; Scherrer and Erjavec 2013; Pettersson, Megyesi, and Nivre 2014). CSMT models have been shown to outperform stochastic transducers on a number of tasks (Tiedemann and Nabende 2009); they are more flexible as phrases can be long (up to 10 characters) and of variable length.

Existing SMT toolkits can be used for CSMT with a simple trick: spaces are inserted between the characters of a word, so that each character is interpreted as a word, and the whole word as a sentence. “Real” inter-word spaces could be converted to a special symbol (e.g., an underscore character) before inserting the spaces between characters, but we do not make use of this possibility in this paper since we only translate single words.

2.3 Supervised and unsupervised modernisation with CSMT

We present two experiments in this paper. In the first experiment – a supervised setting – we build a CSMT system analogously to previous work such as Pettersson, Megyesi, and Tiedemann (2013) or Sánchez-Martínez et al. (2013), assuming that training word pairs are available. In the second experiment – a setting which we call unsupervised – we only rely on monolingual word lists (i.e., no word pairs) for training. The first step of this experiment is equivalent to the unsupervised method of Pettersson, Megyesi, and Nivre (2013). However, we do not use the output of this method directly, but rather use it as (noisy) training data to build a CSMT system as a second step. Thus, the training of the CSMT model is not strictly unsupervised since it continues to use supervision in the form of historical word–modern word pairs. We nevertheless prefer to call this experiment unsupervised since, as a whole, it does not require any manually labelled data.

In both experiments we optionally make use of a lexicon of modern words to filter the modernisation hypotheses.

3 The dataset

In this section we introduce the dataset used in the experiments, which is also an independent contribution of the paper. It is freely available and can serve other researchers for experiments on word normalisation while also being comprehensive enough to enable the construction of realistic and useful systems for the modernisation of Slovene.[1] The dataset consists of a training and a testing lexicon of historical Slovene as well as a frequency-annotated reference word list of modern Slovene. In this section we first give a brief introduction of historical Slovene, present the historical corpora from which the lexicons are extracted, the lexicons themselves, and the lexicon of modern Slovene.

3.1 Historical Slovene

Slovene is a South-Slavic language and is, similarly to other Slavic languages, highly inflected. For instance, it still retains the dual number with morphologically distinct forms. In contrast to some other European languages, the orthography of Slovene has been standardised largely only towards the end of the 19th century, meaning that even relatively recent texts exhibit significant orthographic differences compared to the modern standard.

The modern-day alphabet, the so-called Gaj alphabet, was introduced in the 1840s; before that, the Bohorič alphabet, modelled on the German one, was used. The difference between the two concerns six sounds (IPA *ts*, *s*, *z* and their palatalised variants), with the standard mapping from Bohorič to Gaj (although not always strictly observed, specially in older texts) being, for lower-case letters: z → c, ſ → s, s → z, zh → č, ſh → š, sh → ž. All the letters of the old alphabet, except ſ, are still used today but they correspond to different sounds, which makes reading texts in the Bohorič alphabet difficult and, to some extent, also complicates identifying the alphabet used.

We have split the historical texts into three slices, each covering approximately a 50-year period. The latter two are also split according to the alphabet they use:

18B Texts from the second half of the 18th century, all written in the Bohorič alphabet;
19A Texts from the 19th century (mostly its first half) written in the Bohorič alphabet;
19B Texts from the 19th century (mostly its second half) written in the Gaj alphabet.

As an example of the kinds of texts and modernisations we encounter in historical Slovene texts, we give text snippets from the three slices in Figure 1. All snippets contain the word *ljubezen* "love". As can be seen, the differences are not only due to the change in the alphabet but also encompass other spelling changes, such as *prut → proti*, due to phonological, morphological or orthographic changes or differences between authors.

[1] The dataset is available under CC BY (for the historical lexicons) and CC BY-NC-SA (for the lexicon of modern Slovene) from `http://nl.ijs.si/imp/experiments/jnle-dataset/`.

18B	Al	ta	nar bòl	vashna	reſsniza	je	moja	lubęsen	prut	Nęshki.		
(1790)	*ali*	*ta*	*najbolj*	*važna*	*resnica*	*je*	*moja*	*ljubezen*	*proti*	*nežki*		
19A	poboshnim	ſerzam	in	veſtjo	pridnoſt	in	ljubesin	k	ſvojimu	ſtanu	sdrushi	
(1843)	*pobožnim*	*srcem*	*in*	*vestjo*	*pridnost*	*in*	*ljubezen*	*k*	*svojemu*	*stanu*	*združi*	
19B	Otroška	ljubezen	naj	zmír	te	navdaja	Za	starše,	za	brate,	Bogá	in	cesarja
(1872)	*otroška*	*ljubezen*	*naj*	*zmeraj*	*te*	*navdaja*	*za*	*starše,*	*za*	*brate,*	*boga*	*in*	*cesarja*

Fig. 1. Slovene text from three different periods. The column in bold shows the slice the text belongs to and, in brackets, its year of publication. Each example gives the original text in the first line and the modernised word tokens in the second line, to illustrate the kind of phenomena that must be handled in the modernisation of words.

3.2 Corpora of historical Slovene

The data used in the experiments comes from the IMP resources of historical Slovene (Erjavec 2015).[2] In the experiments we used (lexicons derived from) two corpora called *goo* and *foo*, which are based, respectively, on the IMP goo300k corpus, comprising about three hundred thousand words, a previous version of which is described in Erjavec (2012), and IMP foo3M, a three million word corpus. The *goo* and *foo* corpora are almost identical to the originals, only slightly smaller, as four outlier texts have been removed: two books containing highly idiosyncratic ways of spelling words and two small samples of much older texts. The *goo* corpus contains individual sampled pages from historical Slovene texts and was fully manually annotated in several annotation campaigns. The *foo* corpus contains further sampled pages of the texts included in *goo* as well as from additional texts. In contrast to *goo*, the *foo* corpus is only partially manually annotated: it was built with the purpose of extending the lexicon of historical Slovene, so words already covered by *goo* were not manually annotated.

The key figures of both corpora are given in Table 1, per period and in total. Each text corresponds to a book or a newspaper issue, while pages are, as mentioned, the unit of sampling. The column headed "Words" is the number of all word tokens, while "Verified" gives the number of manually verified word tokens and their annotations, in particular the modernised form of the word. It should be noted that we did not take into account verified multi-word tokens, which is why the "Verified" column is, for the fully manually annotated *goo*, slightly smaller than the complete number of words. In *foo* we further discounted some tokens which had been verified (because of over-enthusiastic annotators) even though these word forms already appear in *goo*. The last line gives the total for each corpus, where it should be noted that the totals can be smaller than the sum of the slices, since texts and pages (but not tokens) may overlap between the slices, as a few are written in both alphabets and can therefore belong to both 19A and 19B.

The text in the two corpora is tokenised, and each token is annotated with its modernised form, its lemma, part-of-speech tag, and, for archaic words, its gloss containing synonym(s)

[2] The IMP resources are available from `http://nl.ijs.si/imp/`.

Table 1. *Sizes of the goo and foo corpora in terms of texts, pages and (verified) words.*

	goo corpus				*foo* corpus			
	Texts	Pages	Words	Verified	Texts	Pages	Words	Verified
18B	8	155	22,100	21,807	11	1,000	146,060	15,353
19A	9	122	41,861	41,468	18	697	401,423	14,682
19B	70	751	203,163	202,020	297	2,873	2,358,792	66,393
Σ	85	1,015	267,124	265,295	321	4,500	2,906,275	96,428

from modern Slovene or a short explanation of its meaning. Archaic words were taken to be those which are not used anymore in modern Slovene, exhibit a significant semantic shift compared with current usage or have changed their lexical morphosyntactic properties, e.g., now have a different gender.

For the presented experiments, the most important information is the original word form and its modernised form, which is the form of the word as it is (or would be, for archaic words) written today: the task of the experiments will be to predict the correct modernised form given the word form. So, for example, a correct mapping for slice 19A in Figure 1 would be *ljubesin* → *ljubezen*.

3.3 Lexicons of historical Slovene

From the verified tokens of the two corpora we have extracted the training and testing lexicons, used in the subsequent experiments. The lexicon extracted from the *goo* corpus, called L_{goo}, will be the training lexicon, as it represents high-frequency words and covers all parts of speech. The lexicon extracted from the manually verified word tokens in the *foo* corpus, called L_{foo}, will be the testing lexicon. This separation into training and testing lexicons gives a very good approximation of the kind of out-of-vocabulary (OOV) words that the system would have to deal with in a real-world setting, as L_{foo} contains only lower frequency words, i.e., those not already seen in the basic L_{goo} lexicon.

In constructing the lexicons we filtered out words which are out of scope for historical word modernisation and its evaluation: digits, foreign words, typos, individual left-over Bohorič words in 19B and Gaj words in 19A, and cases where one word modernises to several words or vice-versa. We also removed from L_{foo} words which already appear in L_{goo}, making the two lexicons disjoint in word forms.

Each lexical entry consists of a triplet ⟨*wform*, *nform*, *mform*⟩, where:

- *wform* is the historical word form as it appears in the corpus, only lower-cased;
- *nform* is the trivially normalised form of *wform*: it is identical to *wform*, except that Bohorič spellings are transliterated to Gaj and that vowel diacritics, which hardly ever appear in modern Slovene, are removed; the *nforms* will be used as one of two baselines for our experiments (see Section 4.1);
- *mform* is the modernised form of *wform*, as it was manually annotated in the corpus.

Tables 2 and 3 present the sizes of the two lexicons. Together, they offer a substantial resource for historical Slovene, as they contain over 60,000 word forms, corresponding to

Table 2. *Properties of the training lexicon* L_{goo}.

Period	Entries	Unique *wforms*		Unique *nforms*		Unique *mforms*		*wform=mform*		*nform=mform*	
18B	6,644	6,494	97.7%	6,019	90.6%	5,065	76.2%	1,181	17.8%	2,854	43.0%
19A	11,600	11,352	97.9%	10,250	88.4%	9,594	82.7%	2,755	23.8%	7,912	68.2%
19B	28,011	27,252	97.3%	26,084	93.1%	23,888	85.3%	19,635	70.1%	21,112	75.4%
Σ	41,915	40,688	97.1%	35,609	85.0%	30,630	73.1%	20,825	49.7%	28,419	67.8%

Table 3. *Properties of the testing lexicon* L_{foo}.

Period	Entries	Unique *wforms*		Unique *nforms*		Unique *mforms*		*wform=mform*		*nform=mform*	
18B	4,774	4,641	97.2%	4,121	86.3%	3,685	77.2%	340	7.1%	1,232	25.8%
19A	5,907	5,801	98.2%	4,942	83.7%	4,830	81.8%	890	15.1%	3,613	61.2%
19B	10,673	10,470	98.1%	9,974	93.5%	9,826	92.1%	8,120	76.1%	8,595	80.5%
Σ	20,569	20,077	97.6%	17,765	86.4%	16,393	79.7%	8,948	43.5%	12,877	62.6%

over 45,000 modern forms. In both datasets the distribution is uneven among the periods, with more data being available for the later periods.

The percentages following the numbers give their ratios against the number of entries in the row. The "Unique *wforms*" percentage shows the inherent ambiguity of the word forms: one *wform* can map to several *mforms*. On average, this ambiguity is somewhat less than 3%, and is highest over the L_{goo} lexicon merged for all slices. This percentage represents the upper bound on precision for our approach, which deterministically modernises isolated word forms.

The ambiguity of *nforms*, represented by the "Unique *nforms*" percentages, is much higher, i.e., over 16% in the worst case (19A in L_{foo}). Thus, advanced modernisation techniques should be applied directly on the *wforms* rather than on the *nforms*.

The "Unique *mforms*" percentages indicate the variability of historical spelling, i.e., how many different entries correspond to one *mform* on average. This variability is greatest in the 18B period, which is to be expected, as standardisation of spelling was then at its weakest. Nevertheless, even in the most recent period (19B in L_{goo}) the variability of spelling is still almost 15%. There is less variability in L_{foo}, presumably because variability is greatest among closed-class words, most of which are already covered in L_{goo}.

Finally, the last two columns show in how many entries the historical spelling is identical to the modern one. These numbers are very low for the Bohorič periods (18B and 19A), which is to be expected, but even in the Gaj period (19B), where the same alphabet is used, the difference is still large, around 25%. In the *nform=mform* column, the percentages rise in the Bohorič periods as a result of alphabet normalisation, e.g., by almost 80% relative in 18B L_{foo}. However, it should also be noted that the normalisation of vowel diacritics still brings some improvement in the 19B period, with about 5% absolute gain. The numbers of

Table 4. *Out-of-vocabulary words in L_{goo} and L_{foo}.*

	L_{goo}					L_{foo}				
	mforms	OOV words		Archaic words		*mforms*	OOV words		Archaic words	
18B	5,065	740	14.6%	507	10.0%	3,685	663	18.0%	414	11.2%
19A	9,594	1,180	12.3%	718	7.5%	4,830	909	18.8%	565	11.7%
19B	23,888	4,417	18.5%	2,139	9.0%	9,826	1,799	18.3%	772	7.9%
Σ	30,630	6,204	20.3%	3,273	10.7%	16,393	3,307	20.2%	1,708	10.4%

this last column for L_{foo} will serve as a baseline precision for the modernisation of Slovene historical words.

3.4 The lexicon of modern Slovene

Most systems for historical word modernisation rely on a lexicon of modern word forms against which the hypotheses are filtered. For our experiments we use Sloleks, an inflectional reference lexicon of modern Slovene,[3] which contains about 100,000 lemmas with their full inflectional paradigms. The word forms are also annotated with their frequency of occurrence in the 1 billion word Gigafida corpus (Logar Berginc, Grčar, Brakus, Erjavec, Arhar Holdt, and Krek 2012).

For the purposes of this experiment, we extracted from Sloleks the list of its lower-cased word forms (930,000) together with their frequencies. We then matched the *mforms* in the two historical lexicons against this list and marked each *mform* which does not appear in Sloleks with a flag in order to have a convenient way of observing the difference in behaviour between in-vocabulary and out-of-vocabulary (OOV) words. We also identify archaic words; in contrast to other OOV words they are, almost by definition, not included in a lexicon of the modern language.

In Table 4 we give the number of OOV modern words and the number of archaic words, also in percentages calculated on the number of modern forms. The number of OOV words is not negligible: overall, every fifth word is missing from Sloleks. This shows that Sloleks, in spite of its size, still has low coverage for the kinds of words used in historical texts, especially for proper nouns. However, between 40 and 70% of OOV words are archaic, and such words will hardly ever appear in a lexicon of modern words. This means that any modernisation approach that filters its hypotheses against such a lexicon will fail in a large proportion of cases.

4 Experiments and results

The data described above allows us to conduct several experiments for the automatic modernisation of historical Slovene words using CSMT. CSMT systems consist of a translation model, which is trained on historical word–modern word pairs, and a language model,

[3] Sloleks is available under the CC BY-NC-SA license from `http://www.slovenscina.eu/`.

which is trained on a list of modern words. We present below two types of experiments that differ in the way the translation model training data is obtained: the supervised experiments (Section 4.2) use the word pairs from L_{goo}, whereas in the unsupervised experiments (Section 4.3) the historical words from L_{goo} are matched automatically with modern word candidates from Sloleks, simulating a scenario where manually annotated modernisations are not available. All models are tested on the word pairs of L_{foo}.

Our experiments have been carried out with the tools of the standard SMT pipeline: GIZA++[4] (Och and Ney 2003) for alignment, Moses[5] (Koehn et al. 2007) for phrase extraction and decoding, and IRSTLM[6] (Federico, Bertoldi, and Cettolo 2008) for language modelling.

Before presenting the experiments we introduce our evaluation metrics and consider two baselines.

4.1 Evaluation and baselines

We evaluate our models on modernisation *accuracy*, defined as the percentage of automatically modernised words that are identical with their manually annotated *mform* in L_{foo}. Accuracy thus indicates only whether the proposed modernisation is correct or not, but in the latter case it does not tell us how incorrect the proposed word is. We therefore add a more precise evaluation measure, *character error rate* (CER), which measures the difference between the proposed word and the gold standard word at the character level. It is defined as the minimal number of edit operations (insertion, deletion or replacement of a character) required to transform the proposed word into the gold standard word, normalised by the length in characters of the latter; hence, lower values are better. In other words, CER is a length-normalised variant of Levenshtein distance.

We will compare the results of our models with two baseline systems (see Table 5). Baseline 1, already introduced in Section 3.1, consists of a set of rules that transform the Bohorič alphabet to the modern one and remove vowel diacritics; the accuracy of Baseline 1 corresponds to the figures of the *nform=mform* column of Table 3. Of course, this baseline does not address all the phenomena in language change from historical to modern Slovene, but only those that are the most simple to implement, while at the same time covering many words.

Baseline 2 corresponds to a basic variant of the modernisation system of Pettersson, Megyesi, and Nivre (2013). For each historical word, we search for the most similar modern word available in Sloleks in terms of Levenshtein distance. If several candidates with the same distance are found, we select the one for which Sloleks shows the highest frequency of occurrence. For performance reasons, we adopt one single parameter proposed by Pettersson, Megyesi, and Nivre (2013): the potential modernised words must be at most four characters shorter or one character longer than the historical word. In order to keep

[4] `https://code.google.com/p/giza-pp/`

[5] `http://www.statmt.org/moses/`

[6] `http://hlt.fbk.eu/technologies/irstlm-irst-language-modelling-toolkit`

Table 5. *Baseline and upper bound performances on L_{foo}. Corr. stands for the absolute numbers of correct entries, Acc. for Accuracy and CER for character error rate.*

	Total	Baseline 1			Baseline 2			Upper bound	
	entries	Corr.	Acc.	CER	Corr.	Acc.	CER	Corr.	Acc.
18B	4,774	1,232	25.8%	0.194	1,554	32.6%	0.273	4,641	97.2%
19A	5,907	3,613	61.2%	0.082	2,653	44.9%	0.196	5,801	98.2%
19B	10,673	8,595	80.5%	0.040	8,165	76.5%	0.061	10,470	98.1%

the baseline as language-independent as possible, we do not take into account other proposed parameters such as edit distance thresholds, context-sensitive edit distance, compound splitting and existing manual translations.

The baseline figures in Table 5 show how the language is coming closer to the modern standard: while only one fifth to one third of words from 18B are easily modernised, this proportion rises to three quarters in 19B; for all slices, there is still room for improvement. For the 19A and 19B slices, a small language-specific set of transformation rules (Baseline 1) works better than a language-independent unsupervised normalisation method (Baseline 2).

Our CSMT models operate deterministically, which means that ambiguous *wforms* (i.e., historical forms with more than one possible modernisation) will be associated with a single *nform*. Hence, the number of unique *wforms* represents the upper bound of our models. Upper bound values are also reported in Table 5.

4.2 Supervised learning

4.2.1 Experimental set-up

In this section, we detail the training data used for the supervised experiments and the parameter settings for Moses that we have found to work best on our data.

Translation model training data. We train four models, three for each slice (18B, 19A, 19B), and a fourth model combining the data from all slices.[7] The combined model simply uses the union of training data of the three slices, which means that the model is biased towards 19B, since most training data comes from this slice. Still, this setting is probably the most realistic one when fine-grained temporal annotations are not available. The goal of splitting the experiments into the three slices is also to determine whether it is better to have smaller but more precise models or a larger but noisier one. In all experiments we keep the OOV entries as well as the archaic entries. While excluding such entries from the data may result in cleaner and better performing models, such annotations may not be available in real-world settings.

[7] The training data sizes thus correspond to the “Entries” column of Table 2.

Language model training data. This data is composed of the Sloleks word forms. While we could have complemented the language model with the modernised OOV forms of L_{goo}, we chose not to do so to keep the setting comparable with the unsupervised experiment. In any case, such an addition would not have a direct impact on the results since the test lexicon L_{foo} is disjoint from the training lexicon L_{goo}. Each word of the lexicon is entered into the language model once; repeating the words according to their frequency (which simulates training directly on a corpus) did not improve the results.

Language model order. This represents the maximum number of consecutive characters that are modelled. Higher orders define the context more accurately and allow us to model entire words. However, this comes at the cost of higher data requirements, so that for small datasets, lower orders may be more successful. We have obtained the best results with a 5-gram language model, which seems to be able to capture sufficiently large contexts while remaining compact.

Character alignment. In order to create a CSMT translation model, the training word pairs need to be aligned character by character. While this can be done using weighted finite state transducers (Jiampojamarn, Kondrak, and Sherif 2007) or using a simple method based on the longest common subsequence (Sánchez-Martínez et al. 2013), better results have been obtained with GIZA++, a more complex tool originally developed for aligning words in parallel sentences (Pettersson, Megyesi, and Tiedemann 2013; Pettersson et al. 2014).

One major drawback of using GIZA++ for character alignment is its excessive reliance on context-independent co-occurrence statistics. In other words, GIZA++ is often not able to distinguish different character mappings according to their context (Tiedemann 2012). To include additional contextual information in the alignment process, we followed Tiedemann (2012) and converted the character sequences into bigram sequences, e.g., *abc* was tokenised as *ab bc c_*. After the alignment process the bigram sequences were converted back to character sequences. It should be noted that we also experimented with keeping bigram representations throughout the model, which typically increases the model size, but we have obtained accuracy improvements only for the 19B slice where most data is available.

Alignment combination and phrase extraction. The alignment process is done in both directions (from historical to modern Slovene and from modern to historical Slovene) in order to create a phrase-based translation model. It would have been most intuitive to align the words only in the direction of historical evolution, i.e., from historical to modern Slovene, but this would have prevented us from obtaining many-to-one alignments, which do exist in our data. Phrase pairs are extracted and scored using the default procedures and settings of the Moses toolkit, i.e., using the *grow-diag-final* heuristic for alignment combination and setting the maximum phrase length to 7 characters.

Smoothing. It has been found that the probability of alignments with small absolute frequencies are typically overestimated, and this effect is countered by applying *Good-Turing discounting*. This consistently improved accuracy by 0.5% absolute on average.

Distortion. Distortion is commonly used in word-level SMT to account for the fact that different languages have different word orders. It tells the models which words are likely to change position in which contexts. In CSMT this setting is usually disabled since the equivalent character-level phenomenon, metathesis, is very rare in the modelled language varieties. We also disabled distortion, since there is not much evidence of metathesis in the historical evolution of Slovene.

Tuning. In current SMT models, the score of a translation is a log-linear combination of scores coming from different features. Depending on the setting, some of these features may be more important than others. The weights of these features are adjusted iteratively by translating some unseen tuning data and evaluating its correctness. The most popular flavour of this process is known as Minimum Error Rate Training (MERT) (Och 2003). We used MERT with the default optimisation objective, i.e., BLEU score over (character) 4-grams. While Tiedemann (2012) has argued that word-level BLEU is preferrable, this is only feasible if the material to be translated actually consists of more than one word at the time; when translating single words, word-level BLEU amounts to a binary true/false measure which is even less fine-grained than character-level BLEU. However, using higher-order BLEU (e.g., 10-grams instead of 4-grams) could further improve accuracy – this remains subject to further research.

We set every tenth word of the training set aside for MERT tuning and used the remaining 90% for alignment and phrase extraction. This setup gave us accuracy gains of up to 15% (relative) compared to a configuration where default weights were used instead of MERT, and the entire training set was used for alignment and phrase extraction. Due to space limitations, we can merely give an example of the weights obtained by a typical MERT run: for a model trained on all slices, we obtain a language model weight of 0.05, inverse phrase translation and lexical weights of 0.01 and 0.05, and direct phrase translation and lexical weights of 0.38 and -0.06. In all settings, the direct phrase translation probability is found to be the most important feature.

Since MERT is non-deterministic, every run yields slightly different results. The reported results represent majority votes over three MERT runs on the same 90%-10% data split. We never had to resort to random selection of answers, since in all experiments and for all words, at least two of the three results agreed.

Lexicon filter. The candidates proposed by the CSMT system are not necessarily existing modern Slovene words. Following Vilar et al. (2007), we added a lexicon filter after the language model to favour existing words over non-words. In practice, we generated 50-best candidate lists with Moses, and selected the first candidate that also occurs in Sloleks. In case none of the 50 candidates occurs in Sloleks, we returned the candidate with the best Moses score. We report results with and without lexicon filter.

4.2.2 Results

We evaluate the CSMT modernisation models on the L_{foo} lexicon including OOV and archaic words, since a fair test should not know whether the modernised word is in the modern lexicon or not.

Table 6. *Results of the supervised experiments on L_{foo}, in terms of absolute numbers of correct entries (Corr.), accuracy (Acc.) and character error rate (CER). The best results for each slice are given in bold.*

	Period	Total	Without lexicon filter			With lexicon filter		
		entries	Corr.	Acc.	CER	Corr.	Acc.	CER
Trained	18B	4,774	2,929	61.4%	0.101	3,236	67.8%	0.095
on single	19A	5,907	4,414	74.7%	0.056	4,633	78.4%	0.056
slices	19B	10,673	9,246	**86.6%**	**0.030**	9,034	84.6%	0.040
Trained	18B	4,774	2,603	54.5%	0.111	3,269	**68.5%**	**0.090**
on all	19A	5,907	4,471	75.7%	**0.054**	4,684	**79.3%**	**0.054**
slices	19B	10,673	9,230	86.5%	0.031	8,985	84.2%	0.041

The results are shown in Table 6. We use test data from the three slices to evaluate the single slice model from the corresponding slice (upper rows) and to evaluate the all slice model (lower rows). All CSMT models beat both baselines for all periods, in terms of accuracy as well as character error rate. While accuracy almost doubles in the 18B slice, the gains decrease as the language becomes closer to the modern one.

The question whether it is better to train the models on a small dataset of temporally precise texts or on a large dataset ranging over several time periods is difficult to answer: among the six experiments, a statistically significant difference could be obtained only for the 18B experiment without lexicon filter, in favour of the single slice model (chi-square tests on accuracy at 95% confidence level).

In an additional experiment, when testing on data that is not annotated with time slices (i.e., testing on the union of 18B, 19A and 19B), the model trained on all slices unsurprisingly performed better than any of the single slice models.

We manually evaluated a random set of 150 (50 per slice) incorrect results produced by the supervised experiment without the lexicon filter in order to determine what kind of errors the CSMT system makes. We classified these proposed modernisations into those that either are word forms of modern Slovene (but incorrect ones) or could be word forms, given the spelling rules of modern Slovene (but are not), and into those that do not conform to spelling (and phonological) rules of Slovene. For example, for the historical word form *uplojitve* the suggested modernisation was *uplojitve* (i.e., the same form), and this word form could well be a modern one; however, the correct modernisation is *oploditve*. On the other hand, the suggested modernisation of the historical word *delovz* is *delovc*, and modern Slovene spelling rules do not allow a word to have the ending *-vc*: the correct modernisation is, in fact, *delavec*. The evaluation showed that only 12 word forms (8%) of the validated sample break modern Slovene spelling rules; of these 8 were from 18B, 3 from 19A and 1 from 19B. These results show that, by and large, the system produces actual or possible Slovene words, and it is only the lexicon (and possibly the context) that can help the system to prefer one modernisation over another. It also shows, unsurprisingly,

Table 7. *Results of the supervised experiments on the OOV words contained in L_{foo}.*

	Period	OOV words		Without lexicon filter		With lexicon filter	
		Total	% of all	Correct	Accuracy	Correct	Accuracy
Trained	18B	663	13.9%	324	48.9%	203	30.6%
on single	19A	909	15.4%	631	69.4%	417	45.9%
slices	19B	1,799	16.9%	1,512	84.1%	843	46.9%

that more ill-formed modernisations are produced for the earlier texts, as these are more different from today's standard than are the later ones.

For the experiments with the lexicon filter we have found, in contrast to Vilar et al. (2007), that it is helpful for the earlier slices. In 18B, adding the lexicon filter improves accuracy by up to 25% relative, and CER values improve accordingly. In 19A, accuracy improves by a few percent, but CER values are unchanged. In 19B, the lexicon filter degrades the output. One reason for this result is that the relative importance of OOV words becomes higher in later slices. Thus, the lexicon filter removes some good candidates just because they are not in Sloleks. In contrast to the earlier periods, this negative effect is not balanced out by enough positive answers in 19B.

It is thus interesting to see how our models cope with OOV words, i.e., words that are unknown to the language model and the lexicon filter. The results on just the OOV entries of L_{foo} are given in Table 7. Indeed, the models with lexicon filter perform consistently worse on these entries: not only will they never find the correct word in the lexicon, but they are likely to boost incorrect candidates just because they are available in the lexicon. This experiment highlights the importance of the lexicon filter: if it has insufficient coverage, it is better not to employ it, as any gain of using the filter will be offset by the degradation of precision with OOV words.

The transformation rules of Baseline 1 concern the "easy cases" of simple and regular spelling differences. As a sanity check, it is worth seeing how many of these easy cases the CSMT models are able to transform correctly. The models trained on a single slice, used without the lexicon filter and evaluated on the *mform=nform* entries show accuracy values of 87.7% (18B), 97.0% (19A) and 97.8% (19B), suggesting that they indeed pick up the regular spelling changes in a satisfying way.

4.3 Unsupervised learning

The supervised approach requires training data in the form of word pairs. Owing to its manual annotation the L_{goo} lexicon does contain such pairs of historical words and their modernised counterparts. However, such a resource may not be available for other language varieties. Thus, in this second experiment, we investigate what can be achieved with purely "monolingual" data. Concretely, we start with a list of historical words (extracted from L_{goo}) and a list of modern words (extracted from Sloleks). We propose to use the Baseline 2 model as a bootstrapping step that aligns the words of the two lists based on orthographic similarity, and then train the CSMT system on these hypothesised word pairs.

Table 8. *Results of the unsupervised experiments on L_{foo}, in terms of absolute numbers of correct entries (Corr.), accuracy (Acc.) and character error rate (CER).*

	Period	Total	Without lexicon filter			With lexicon filter		
		entries	Corr.	Acc.	CER	Corr.	Acc.	CER
Trained	18B	4,774	2,017	42.2%	0.190	2,346	**49.1%**	**0.183**
on single	19A	5,907	3,789	64.1%	**0.098**	3,967	**67.2%**	0.099
slices	19B	10,673	8,407	**78.8%**	0.051	8,259	77.4%	0.059
Trained	18B	4,774	1,849	38.7%	0.204	2,218	46.5%	0.194
on all	19A	5,907	3,564	60.3%	0.114	3,821	64.7%	0.109
slices	19B	10,673	8,393	78.6%	**0.050**	8,246	77.3%	0.058

4.3.1 Experimental set-up

The bootstrapping step consists of searching, for each historical word of L_{goo}, its most similar modern word in Sloleks. We use the same matching algorithm as for Baseline 2, i.e., the Levenshtein distance as the similarity measure and modern word frequencies to resolve ties, as well as the mentioned string length restrictions to reduce processing time. While the matching algorithm is applied to L_{foo} for Baseline 2, we apply it to L_{goo} here.

The result of the bootstrapping step is a list of word pairs which is noisier than the one used in the supervised experiment: 45% of 18B word pairs, 55% of 19A word pairs and 75% of 19B word pairs are correct. However, incorrectly induced word pairs are still useful, since they are rarely totally incorrect and can thus serve the CSMT training for the parts that are correct. So, for example, the historical word *zerkovne* is matched to the modern word *cerkovne*, while the correct modernisation is *cerkvene*. The extracted pair thus fails to show the change *ov* → *ve*, but does correctly predict the change *z* → *c*.

We induce word pairs for each of the three periods (18B, 19A, 19B) and for all periods taken together. The word pairs are then used to train CSMT models, using the same settings as those reported in Section 4.2, except for MERT. Tuning with MERT deteriorated the results drastically on most conditions, because the tuning data is partially incorrect (it is also induced in the bootstrapping step). Interestingly, the ranking of the different features induced with MERT on the unsupervised models is very similar to the one induced in the supervised setting. For consistency, all the results reported here are without MERT.

4.3.2 Results

Again, we conducted experiments for the three time slices. We tested the system on the word pairs of the L_{foo} lexicon, as above. Results are shown in Table 8.

In terms of accuracy, the CSMT models perform better than Baseline 1 (cf. Table 5) on the 18B and 19A periods. In contrast, Baseline 1 has consistently lower character error rates than the CSMT models. This means that Baseline 1 generates fewer correct modernisations than unsupervised CSMT, but the modernisations it generates are on average closer to the truth than those generated by CSMT.

It is particularly interesting to see whether our two-step approach to unsupervised modernisation outperforms the one-step approach of Pettersson, Megyesi, and Nivre (2013) consisting only of Baseline 2. Our approach beats Baseline 2 by up to 16% for 18B, by up to 22% for 19A, and still by 2% for 19B. The results thus show that the added CSMT step is able to generalise successfully from noisy input data.

As with the supervised experiment, we have also made an evaluation of 150 (50 per slice) randomly selected incorrect results produced without the lexicon filter. This evaluation surprisingly showed better results than the supervised one: only 1 word (compared to 12 in the supervised experiment) breaks modern Slovene spelling rules. This result could be a chance configuration of the examples looked at, but does strongly indicate that the unsupervised system produces actual or possible Slovene words.

Comparing the various CSMT experiments among themselves, the lexicon filter proves useful except for the 19B period, for the same reasons as in the supervised approach. The single slice models outperform the all slice models more clearly than in the supervised approach.

The unsupervised CSMT models should also be able to learn the simple and regular spelling changes implemented in Baseline 1. The models trained on a single slice, used without the lexicon filter, and evaluated on the *mform=nform* entries show accuracy values of 75.7% (18B), 87.2% (19A) and 91.0% (19B): these values are about 10% lower than those obtained in the supervised setting, but they still show acceptable performance on this type of spelling changes.

Tiedemann and Nakov (2013) show that the performance of a CSMT system may be improved by filtering out phrase table entries that are likely to be noise, on the basis of the method presented by Johnson, Martin, Foster, and Kuhn (2007). We expected this additional step to improve the unsupervised model, since it is particularly prone to noisy input data. However, in our case, phrase table filtering lowered accuracy by several percentage points. This negative result is probably due to the small size of the training data, which is several orders of magnitude smaller than the dataset used by Tiedemann and Nakov (2013), so that the entropy statistics used for filtering become less reliable. Our phrase tables were cut by 1/4 – 1/3 of the original size, compared to 1/5 in the cited experiment.

While the unsupervised approach unsurprisingly performs less well than the supervised approach, it is still able to successfully modernise a large number of historical words. Hence, such a model could be used profitably for the initial modernisation of a new dataset, by reducing the amount of manual corrections needed.

4.4 Cross-language comparisons

In this section, we compare our results with related work on automatic modernisation of historical words. Table 9 shows results reported by Pettersson et al. (2014) on English, German, Hungarian, Icelandic and Swedish, and results by Sánchez-Martínez et al. (2013) on Spanish, in addition to our results on Slovene. However, the experimental setups are fairly difficult to compare. For instance, Pettersson et al. (2014) modernise complete texts; in consequence, training and test data are not completely disjoint, and the evaluation metrics refer to tokens. In contrast, Sánchez-Martínez et al. (2013) and our own experiments operate on word types. Furthermore, both related studies create a single model per lan-

Table 9. *Comparison of several modernisation methods on several historical language datasets. Results are given in terms of accuracy (Acc.) and character error rate (CER) where available.*

Language	Period	Identical		Edit distance		CSMT sup.		CSMT unsup.	
		Acc.	CER	Acc.	CER	Acc.	CER	Acc.	CER
English	14th-17th c.	75.8%	0.26	82.9%	0.19	94.3%	0.07	—	—
German	17th-18th c.	84.4%	0.16	87.3%	0.13	96.6%	0.07	—	—
Hungarian	15th-16th c.	17.1%	0.85	31.7%	0.71	80.1%	0.21	—	—
Icelandic	15th c.	50.5%	0.51	67.3%	0.35	71.8%	0.30	—	—
Swedish	16th-19th c.	64.6%	0.36	79.4%	0.22	92.9%	0.07	—	—
Spanish	15th-20th c.	—	0.058	—	0.059	—	0.002	—	—
Slovene 18B	18th c.	7.1%	0.342	32.6%	0.273	68.5%	0.090	49.1%	0.183
Slovene 19A	19th c.	15.1%	0.269	44.9%	0.196	79.3%	0.054	67.2%	0.099
Slovene 19B	19th c.	76.1%	0.049	76.5%	0.061	86.6%	0.030	78.8%	0.051

guage covering very large time spans. Our results on Slovene, however, show that different periods have different spelling particularities that are best accounted for using different models.

The "Identical" column shows how many words or characters can be kept identical during the modernisation process. This column corresponds to the *Baseline* figures in Pettersson et al. (2014), to the *No modernisation* figure in Sánchez-Martínez et al. (2013), and to the *wform=nform* column in our Table 3. In terms of accuracy, the Slovene 18B slice turns out to be the most difficult to modernise: only 7.1% of words are already in their expected modern form. In terms of CER however, the Slovene data shows values comparable with the other languages. In contrast, the Spanish data is already very similar to the modernised forms, probably due to the fact that the Spanish corpus also contains modern-day language – not unlike our 19B slice.

The "Edit distance" column subsumes several comparable modernisation techniques based on an edit distance measure and a modern lexicon filter. It corresponds to our Baseline 2, to some similar experiments but with language-specific thresholds for Pettersson et al. (2014), and to a comparable approach based on a spellchecker for Sánchez-Martínez et al. (2013). Again, this method shows comparatively low accuracy values for Slovene, but similar CER values. The relative improvement with respect to the "Identical" baseline is comparable as well. This method only yields minor improvements for Spanish.

The "CSMT sup." column shows the results of CSMT using gold standard training data, i.e., our supervised setting. We show the best values of all *giza* or *m2m* experiments of Pettersson et al. (2014), the *SMT* experiment of Sánchez-Martínez et al. (2013), and the best setting in terms of accuracy as given in Table 6. Again, the Slovene results show similar patterns as above: comparatively low accuracy values but similar CER values. For all languages, CSMT outperforms the "Identical" baseline as well as the edit-distance-based methods.

Finally, we added the figures for our unsupervised CSMT experiment, where we give the

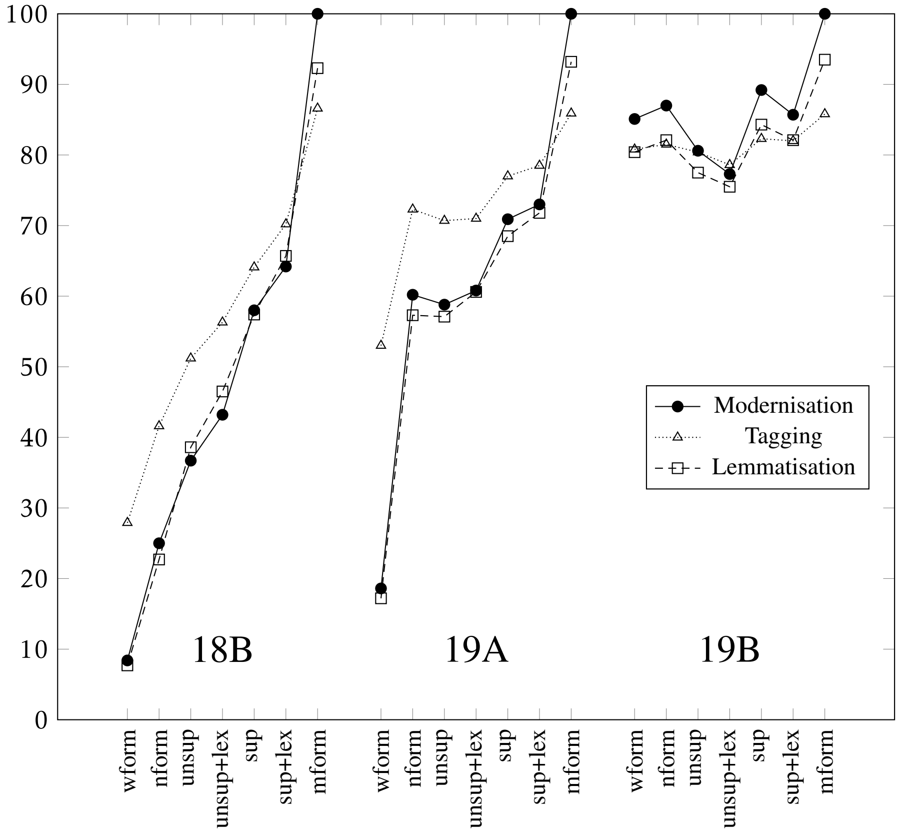

Fig. 2. Accuracies of modernisation, tagging and lemmatisation in the three time slices over the manually validated word tokens of the *foo* corpus. The tokens passed to the tagger and lemmatiser are: *wform*, the lower-cased historical word form from the corpus; *nform*, the Baseline 1 normalisation; *unsup*, the unsupervised modernisation trained on all slices without the lexicon filter; *unsup+lex*, same, but with lexicon filter; *sup*, the supervised modernisation trained on all slices without the lexicon filter; *sup+lex*, same, but with lexicon filter; and *mform*, the manually modernised word form.

best setting in terms of accuracy from Table 8. Such an experiment has not been attempted in previous work; its performance is located about half-way between the edit-distance-based models and the supervised CSMT models.

5 Extrinsic evaluation on PoS tagging and lemmatisation

In order to test the benefits of the presented modernisation techniques on subsequent steps of a typical NLP pipeline, we performed an experiment on part-of-speech tagging and lemmatisation. These tasks are quite difficult even for modern standardised Slovene, which is, as most Slavic languages, highly inflected: the Slovene tagset of morphosyntactic descriptions (MSDs) has almost two thousand distinct tags, while the complicated system of endings and stem alternations, dependent on morphological and syntactic features of a word, makes learning Slovene inflections one of the more daunting tasks for foreign speakers.

We performed the experiment on the *foo* corpus: the complete corpus was tagged and lemmatised, but the results were evaluated only on the manually validated word tokens, i.e., the size of the test set corresponds to the numbers of verified word tokens given for the *foo* corpus in Table 1. The tagging was performed with TnT (Brants 2000) and the lemmatisation with CLOG (Erjavec and Džeroski 2004); the tagging is performed first since the lemmatiser makes use of the MSD tags of the word forms to determine their lemmas. The models for both steps were trained on a manually annotated corpus of modern Slovene, as described in Erjavec, Ignat, Pouliquen, and Steinberger (2005). It should be noted that tagging is performed with the fine-grained MSD tagset, while the manual annotations in the *foo* corpus use the coarse-grained IMP MSD tagset. For tagging evaluation we reduced the fine-grained to the coarse-grained tagset.

The results in terms of per-token accuracy are given in Figure 2. As usual, the data is split according to the time slice, and then accuracies are given separately for normalisation, tagging and lemmatisation. The difference between the settings lies in the word tokens that are passed to the tagger and lemmatiser, starting from the lower-bound *wform*, as it appears in the corpus, and ending with the upper-bound manually modernised *mforms*. In the CSMT settings, we have used the models trained on all slices, even though in some cases the per-slice trained models outperform them. But, as this does not happen in all cases and differences between the settings are slight, it was simpler to adopt one training set for all experiments.

Figure 2 shows that the results of tagging and lemmatisation are quite strongly correlated with the quality of modernisation – the better the modernisation, the better the tagging and lemmatisation results. Somewhat surprisingly, tagging (in 18B and 19A) and even lemmatisation (in 18B) can be more accurate than modernisation. For tagging, the reason lies in the unknown word guessing module of TnT, which is able to predict the correct tag even in cases when the stem (but typically not the ending) of the word has changed. For lemmatisation, the reason is that modernisation at times returns the wrong word form but of the correct lemma, so the lemmatiser still returns the correct result. The second point to note is that the accuracy of tagging and lemmatisation using the correct *mform* is hardly affected by the age of the text, i.e., changes in syntax seem not to have been very large in the 150 years that our data spans, at least to the the extent that this would adversely affect the results.

As already discussed, and confirmed by the tagging and lemmatisation experiments, each of the modernisation methods in 18B as given in the sequence in Figure 2 gives better results than the one preceding it. In 19A the unsupervised method without the lexicon filter is worse than the baseline, and is only slightly better with the lexicon filter. On the other hand, the supervised method performs much better than the baseline, although the lexicon filter only raises the accuracy by a few percent. The 19B slice again confirms previous results: with tagging, only the supervised method without the lexicon filter beats the baseline, and even here by less than 1%. This is also the only method to beat the baseline in lemmatisation, although here the difference is greater, about 4%. In short, the extrinsic evaluation confirms that modernisation does help in obtaining better tagging and lemmatisation of historical texts, but how much the results improve and which method performs best heavily depends on the time period of the text in question.

6 Conclusion

We have applied character-level machine translation to modernise historical Slovene words, obtaining, with the supervised approach, accuracy improvements of up to 35.9% (absolute) and character error rate reductions of up to 0.183 over the Levenshtein distance baseline. With the unsupervised approach, we get accuracy improvements of up to 16.5% (absolute) and character error rate reductions of up to 0.09. The main settings that allowed us to obtain these results are: training on the lexicon rather than directly on the corpus, using a 5-gram language model, using GIZA++ with bigrams to align characters, and disabling distortion. The usefulness of MERT tuning and of training on all slices rather than splitting the training data according to the time period depends on the size and quality of the dataset. For instance, MERT will not perform well in the unsupervised setting, where the tuning data is noisy.

The quality of the results turns out to heavily depend on the time slice of the test data: the results on older data is, of course, worse in absolute terms, but much better in comparison with the baseline. Second, while the lexicon filter generally improves the results, word forms missing from the lexicon degrade the performance. This degradation is most severe in the most recent texts, which are already very similar to modern language.

There are a number of directions we would like our work to take in the future. As noted, our system works on individual word forms and is therefore unable to deal with cases where tokens are merged or split. This is not an infrequent occurrence: in the *goo* corpus, almost 7% of the tokens belong to these types. A simple extension has been proposed by Sánchez-Martínez et al. (2013): they keep the single-word CSMT system, but add 1 to 5 characters of the previous and the following word, so that the modernisation system can take them into account. While this extension has not yielded any benefit to the Spanish data, it could be useful to detect many-to-one word correspondences since it models word boundaries explicitly. Such an extension would also have the advantage to be applicable to the unsupervised setting, whereas the translation of entire sentences with CSMT, as has been done e.g., by Pettersson et al. (2014), is restricted to the supervised scenario. In any case, we do not expect significant benefits from translating entire sentences, since the modernisation of a word rarely depends on more than its immediate context.

Another addition expected to improve accuracy is the inclusion of preliminary part-of-speech tagging, i.e., the opposite setup of Section 5. Here, the hypothesis is that different word classes tend to exhibit different orthographic changes and would benefit from different rules. Moreover, PoS-informed filtering on Sloleks will yield fewer false positives than is currently the case. To accomplish this, we would train a separate CSMT model for each PoS, train a tagger on the *goo* corpus and then use it on the test corpus. The modernisation routine would then use the PoS to select the correct CSMT model to translate the word form. Furthermore, when filtering on Sloleks, we would consider only word forms with the same PoS as the historical word form. Of course, this approach is only possible when there exists a training corpus for historical language, as is the case for *goo*, although with the very coarse IMP tagset.

From a linguistic perspective, it could be interesting to inspect the models (e.g., the CSMT phrase tables) in order to extract the frequency distributions of different phonological and orthographical changes over time, contributing to a better understanding of the diachronic changes in the Slovene language. However, a system that explicitly extracts rules from word pairs (e.g. Bollmann 2012) might be better suited for this specific purpose.

Finally, it would be interesting to extend our approach to other historical languages and other non-standard language varieties.

Acknowledgements

The authors would like to thank Katja Zupan and the five anonymous reviewers for their comments — all errors, of course, remain our own. This work has been partially funded by the LabEx EFL (ANR/CGI), operation LR2.2, by the EU IMPACT project "Improving Access to Text" and the Google Digital Humanities Research Award "Language models for historical Slovenian".

References

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

Beinborn, Lisa, Zesch, Torsten, and Gurevych, Iryna. 2013. Cognate production using character-based machine translation. In *Proceedings of IJCNLP 2013*, pp. 883–91. Nagoya, Japan.

Bollmann, Marcel. 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)*, pp. 3–14. Lisbon, Portugal.

Brants, Thorsten. 2000. TnT – a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP 2000)*, pp. 224–31. Seattle, WA, USA.

De Clercq, Orphée, Desmet, Bart, Schulz, Sarah, Lefever, Els, and Hoste, Véronique. 2013. Normalization of Dutch user-generated content. In *Proceedings of RANLP 2013*, pp. 179–88. Hissar, Bulgaria.

Erjavec, Tomaž. 2011. Automatic linguistic annotation of historical language: ToTrTaLe and XIX century Slovene. In *Proceedings of the 5th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pp. 33–8. Portland, OR, USA.

Erjavec, Tomaž. 2012. The goo300k corpus of historical Slovene. In *Proceedings of LREC 2012*, pp. 2257–60. Istanbul, Turkey.

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

Erjavec, Tomaž, and Džeroski, Sašo. 2004. Machine learning of language structure: Lemmatising unknown Slovene words. *Applied Artificial Intelligence*, *18*(1), 17–41.

Erjavec, Tomaž, Ignat, Camelia, Pouliquen, Bruno, and Steinberger, Ralf. 2005. Massive multilingual corpus compilation: Acquis Communautaire and ToTaLe. In *Proceedings of the 2nd Language and Technology Conference*, pp. 32–6. Poznan, Poland.

Federico, Marcello, Bertoldi, Nicola, and Cettolo, Mauro. 2008. IRSTLM: an open source toolkit for handling large scale language models. In *Proceedings of Interspeech 2008*, pp. 1618–21. Brisbane, Australia.

Fišer, Darja, and Ljubešić, Nikola. 2011. Bilingual lexicon extraction from comparable corpora for closely related languages. In *Proceedings of RANLP 2011*, pp. 125–31. Hissar, Bulgaria.

Fišer, Darja, and Sagot, Benoît. 2015. Constructing a poor man's wordnet in a resource-rich world. *Language Resources and Evaluation*, pp. 1–35.

Jiampojamarn, Sittichai, Kondrak, Grzegorz, and Sherif, Tarek. 2007. Applying many-to-many alignments and Hidden Markov Models to letter-to-phoneme conversion. In *Proceedings of HLT-NAACL 2007*, pp. 372–9. Rochester, NY, USA.

Johnson, Howard, Martin, Joel, Foster, George, and Kuhn, Roland. 2007. Improving translation quality by discarding most of the phrasetable. In *Proceedings of EMNLP-CoNLL 2007*, pp. 967–75. Prague, Czech Republic.

Jurish, Bryan. 2010. Comparing canonicalizations of historical German text. In *Proceedings of the SIGMORPHON 2010 Workshop*, pp. 72–7. Uppsala, Sweden.

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

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

Koehn, Philipp, and Knight, Kevin. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL 2002 Workshop on Unsupervised Lexical Acquisition (SIGLEX 2002)*, pp. 9–16. Philadelphia, USA.

Kondrak, Grzegorz, and Dorr, Bonnie. 2004. Identification of confusable drug names: A new approach and evaluation methodology. In *Proceedings of COLING 2004*, pp. 952–8. Geneva, Switzerland.

Kondrak, Grzegorz, Marcu, Daniel, and Knight, Kevin. 2003. Cognates can improve statistical translation models. In *Proceedings of NAACL-HLT 2003*, pp. 46–8. Edmonton, Canada.

Kondrak, Grzegorz, and Sherif, Tarek. 2006. Evaluation of several phonetic similarity algorithms on the task of cognate identification. In *Proceedings of the ACL 2006 Workshop on Linguistic Distances*, pp. 43–50. Sydney, Australia.

Ljubešić, Nikola, Erjavec, Tomaž, and Fišer, Darja. 2014. Standardizing tweets with character-level machine translation. In *Proceedings of CICLing 2014*, pp. 164–75. Kathmandu, Nepal: Springer.

Logar Berginc, Nataša, Grčar, Miha, Brakus, Marko, Erjavec, Tomaž, Arhar Holdt, Špela, and Krek, Simon. 2012. *Korpusi slovenskega jezika Gigafida, KRES, ccGigafida in ccKRES: gradnja, vsebina, uporaba [The Gigafida, KRES, ccGigafida and ccK-*

RES corpora of Slovene language: compilation, content, use]. Ljubljana, Slovenia: Trojina, zavod za uporabno slovenistiko: Fakulteta za družbene vede.

Mann, Gideon S., and Yarowsky, David. 2001. Multipath translation lexicon induction via bridge languages. In *Proceedings of NAACL 2001*, pp. 151–8. Pittsburgh, PA, USA.

Melamed, I. Dan. 1995. Automatic evaluation and uniform filter cascades for inducing n-best translation lexicons. In *Proceedings of the Third Workshop on Very Large Corpora (WVLC3)*, pp. 184–98. Boston, MA, USA.

Michel, Jean-Baptiste, Shen, Yuan Kui, Aiden, Aviva P., Veres, Adrian, Gray, Matthew K., The Google Books Team, Pickett, Joseph P., Hoiberg, Dale, Clancy, Dan, Norvig, Peter, Orwant, Jon, Pinker, Steven, Nowak, Martin A., and Lieberman Aiden, Erez. 2011. Quantitative analysis of culture using millions of digitized books. *Science*, *331*(6014), 176-82.

Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL 2003*, pp. 160–7. Sapporo, Japan.

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

Pettersson, Eva, Megyesi, Beáta B., and Nivre, Joakim. 2013. 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)*, pp. 163–79. Oslo, Norway.

Pettersson, Eva, Megyesi, Beáta B., and Nivre, Joakim. 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)*, pp. 32–41. Gothenburg, Sweden.

Pettersson, Eva, Megyesi, Beáta B., and Tiedemann, Jörg. 2013. An SMT approach to automatic annotation of historical text. In *Proceedings of the Nodalida Workshop on Computational Historical Linguistics*, pp. 54–69. Oslo, Norway.

Piotrowski, Michael. 2012. *Natural Language Processing for Historical Texts.* Morgan & Claypool.

Rayson, Paul, Archer, Dawn, Baron, Alistair, and Smith, Nick. 2007. Tagging historical corpora – the problem of spelling variation. In *Proceedings of Digital Historical Corpora, Dagstuhl-Seminar 06491.* Wadern, Germany: International Conference and Research Center for Computer Science, Schloss Dagstuhl.

Reffle, Ulrich. 2011. Efficiently generating correction suggestions for garbled tokens of historical language. *Natural Language Engineering*, *17*, 265–82.

Sánchez-Martínez, Felipe, Martínez-Sempere, Isabel, Ivars-Ribes, Xavier, and Carrasco, Rafael C. 2013. *An open diachronic corpus of historical Spanish: annotation criteria and automatic modernisation of spelling*, (Research Report). Alicante: Departament de Llenguatges i Sistemes Informàtics, Universitat d'Alacant.

Scheible, Silke, Whitt, Richard J., Durrell, Martin, and Bennett, Paul. 2011. A gold standard corpus of Early Modern German. In *Proceedings of the 5th Linguistic Annotation Workshop (LAW-V)*, pp. 124–8. Portland, OR, USA.

Scherrer, Yves. 2007. Adaptive string distance measures for bilingual dialect lexicon induction. In *Proceedings of the ACL 2007 Student Research Workshop*, pp. 55–60. Prague, Czech Republic.

Scherrer, Yves, and Erjavec, Tomaž. 2013. Modernizing historical Slovene words with character-based SMT. In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, pp. 58–62. Sofia, Bulgaria.

Scherrer, Yves, and Sagot, Benoît. 2014. A language-independent and fully unsupervised approach to lexicon induction and part-of-speech tagging for closely related languages. In *Proceedings of LREC 2014*, pp. 502–8. Reykjavik, Iceland.

Tiedemann, Jörg. 1999. Automatic construction of weighted string similarity measures. In *Proceedings of EMNLP-VLC 1999*, pp. 213–19. University of Maryland, MD, USA.

Tiedemann, Jörg. 2009. Character-based PSMT for closely related languages. In *Proceedings of EAMT 2009*, pp. 12–19. Barcelona, Spain.

Tiedemann, Jörg. 2012. Character-based pivot translations for under-resourced languages and domains. In *Proceedings of EACL 2012*, pp. 141–51. Avignon, France.

Tiedemann, Jörg, and Nabende, Peter. 2009. Translating transliterations. *International Journal of Computing and ICT Research*, *3*(1), 33–41.

Tiedemann, Jörg, and Nakov, Preslav. 2013. Analyzing the use of character-level translation with sparse and noisy datasets. In *Proceedings of RANLP 2013*, pp. 676–84. Hissar, Bulgaria.

Vilar, David, Peter, Jan-Thorsten, and Ney, Hermann. 2007. Can we translate letters? In *Proceedings of the Second Workshop on Statistical Machine Translation*, pp. 33–9. Prague, Czech Republic.