An SMT Approach to Automatic Annotation of Historical Text

Eva Pettersson[1,2]*, Beáta Megyesi*[1]*, Jörg Tiedemann*[1]

(1) Department of Linguistics and Philology, Uppsala University
(2) Swedish National Graduate School of Language Technology

firstname.lastname@lingfil.uu.se

ABSTRACT

In this paper we propose an approach to tagging and parsing of historical text, using character-based SMT methods for translating the historical spelling to a modern spelling before applying the NLP tools. This way, existing modern taggers and parsers may be used to analyse historical text instead of training new tools specialised in historical language, which might be hard considering the lack of linguistically annotated historical corpora. We show that our approach to spelling normalisation is successful even with small amounts of training data, and that it is generalisable to several languages. For the two languages presented in this paper, the proportion of tokens with a spelling identical to the modern gold standard spelling increases from 64.8% to 83.9%, and from 64.6% to 92.3% respectively, which has a positive impact on subsequent tagging and parsing using modern tools.

KEYWORDS: Digital Humanities, Natural Language Processing, Historical Text, Normalisation, Underresourced Languages, Less-Resource Languages, SMT.

Eva Pettersson, Beáta Megyesi and Jörg Tiedemann 2013. An SMT approach to automatic annotation of historical text. *Proceedings from the workshop on Computational historical linguistics at NoDaLiDa 2013.* NEALT Proceedings Series 18 / Linköping Electronic Conference Proceedings 87: 54–69.

1 Introduction

Historical language may be regarded as an under-resourced language, with limited access to digitized and linguistically annotated corpora, and other NLP resources and tools. Furthermore, it is problematic to develop NLP tools specifically aimed at processing historical text, since the term ”historical” is a wide concept, including texts from a long period of time in which language changes. This means that for example a tagger trained on 14th century text will probably not perform as well on 18th century text. Hence, several taggers would need to be developed for handling different time periods. This problem is further aggravated by the lack of spelling conventions in historical time, meaning that orthography may vary greatly between different authors and genres, or even in the same text written by the same author.

In this paper we propose a method for tagging and parsing of historical text using existing NLP tools developed for modern language. Our approach makes use of standard SMT techniques for normalising the historical spelling to a modern spelling, before applying the NLP tools. Since the machine translation task is performed on a character level, only a small parallel corpus of historical and modern spelling is needed for training.

We show that this method is successful for automatic annotation of historical text, even with small amounts of training data. Furthermore, our two case studies illustrate that this method can be generalised to several languages.

2 SMT-based Tagging and Parsing

The approach to automatic analysis of historical text presented in this paper is illustrated in Figure 1. The input file is a historical text in its original spelling. This text is first tokenised using standard tokenisation. The tokenised text is then normalised to a modern spelling, using character-based SMT, as described further in Section 3. The normalised text is used as input for tagging and parsing, which is performed by available tools for the modern language. In the last step, the annotation is projected back to the original spelling. The final output is thus a tagged and parsed file with the historical spelling preserved.

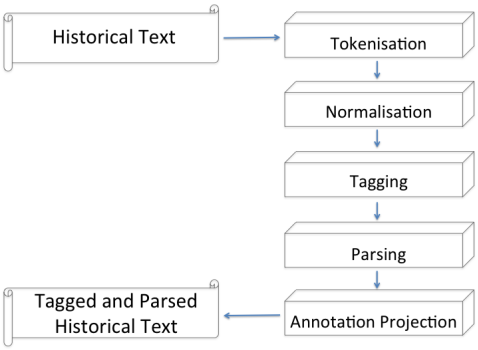

Figure 1: SMT-based tagging and parsing of historical text.

3 Normalisation Using SMT

As stated in Section 1, we regard normalisation of historical text as a translation task. In contrast to traditional translation tasks, normalisation should be performed on a lower level addressing changes in spelling instead of the translation of words and phrases. Hence, treating text normalisation as a case of statistical machine translation leads to a character-level approach without lexical re-ordering. The use of phrase-based SMT in transliteration and character-level translation between closely related languages has already been shown in (Matthews, 2007; Vilar et al., 2007; Nakov and Tiedemann, 2012). We will follow their ideas by applying similar techniques to historical texts.

The basic idea of character-level SMT is that phrases are modeled as character sequences instead of word sequences. Translation models are then trained on character-aligned parallel corpora, and language models on character N-grams. Nakov and Tiedemann (2012) have shown that small parallel training corpora are sufficient for reasonable performance of such an approach. Language models can be trained on larger monolingual corpora, and higher orders in terms of N-gram size can be used to ensure fluent and grammatically correct output (Nakov and Tiedemann, 2012).

The training data in our case is simply a set of word pairs with historical and modern spelling respectively. In contrast to the application of character-level SMT to related language translation, we assume a strict one-to-one correspondence between words of the historical text and its transformation into modern spelling. This simplifies the general setup and ensures that training examples are sufficiently short to enable efficient training procedures.

Important for the success of character-level SMT is the proper alignment of characters that will be used to estimate the parameters of the character-based translation model. We follow the approach of Nakov and Tiedemann (2012) and apply two different tools for this purpose, a weighted finite state transducer implemented in the m2m-aligner (Jiampojamarn et al., 2007), and the word alignment toolkit GIZA++ implementing the IBM models used in statistical MT. Similar to their experiments on related languages, we would like to verify that these alignment techniques are also effective for training spelling normalisation models for historical texts.

The m2m-aligner implements transducer models based on context-independent edit operations. Furthermore, the toolkit allows to include operations over character N-grams (instead of single characters). The transducer is trained using EM on (unaligned) parallel training data. The final model can then be used to produce a Viterbi alignment between given pairs of character strings.

An example is shown in Figure 2, illustrating the alignment of the Icelandic historical spellings *meðr* and *giallda* to the modern versions *meður* and *galda*. In this example, the ϵ symbol denotes empty alignments, i.e. insertions and deletions. For instance, the ϵ symbol in the source word *meðr* denotes the insertion of *u* in the target word *meður*. Likewise, the ϵ symbol in the target word *galda* denotes the deletion of *i* as compared to the source word *giallda*. 2:1 and 1:2 alignments are also possible, as in the case of the alignment of *giallda* to *galda*, where the colon denotes that both letters *l* and *d* in the source word correspond to the single letter *d* in the target word.

m|e|ð|ϵ|r|	m|e|ð|u|r|
g|i|a|l|l:d|a|	g|ϵ|a|l|d|a|

Figure 2: Character level alignment.

Similarly, GIZA++ is used to train alignment parameters on parallel training data using EM. The toolkit supports several word alignment models which are applied in the order of increasing complexity (see Och (2003) for more details). The final model can then be used again to produce a Viterbi alignment over the data. Word alignment models are certainly not developed for character-level alignment and some of their parameters are not suited for this task. Nevertheless, Nakov and Tiedemann (2012) demonstrate the success of these techniques, which also outperform the transducer-based approach. Therefore, we will revisit these two approaches for our purposes.

Finally, Viterbi alignments are used to extract translations of character sequences and their distributions are used to estimate translation model parameters in the same way as it is done for word-level phrase-based SMT.

4 Experimental Setup

We will run our experiments on two separate data sets: one for Icelandic and one for Swedish, as described further in Section 5 and 6 respectively.

In general, we will focus on phrase-based SMT, i.e. a decomposition of translation models into mappings between non-overlapping character sequences. The SMT engine used is Moses with all its standard components. We apply a phrase-based model with the common feature functions: 2 phrase translation probabilities, 2 lexical weights, a phrase penalty feature, a language model feature and a word penalty. The feature weights are trained using MERT with BLEU over character-sequences as the objective function. The phrase table scores are produced using the Moses phrase extraction and scoring methods. The maximum size of a phrase (sequence of characters) is set to 10. Language models (10-gram models) are estimated using SRILM (interpolated and smoothed) which are then transformed into binary versions to be used by KenLM during decoding. Reordering is switched off during decoding (and tuning) as we assume monotonic alignments.

Training character-based SMT models for text normalisation basically involves the following steps (see further Section 3):

- **Word Alignment**
  Aligning corresponding words that will serve as the parallel training data for creating translation models.
- **Character alignment**
  Aligning characters for the entire corpus of aligned word pairs.
- **Language modeling**
  Training character-based language models on monolingual data in the target language.
- **Parameter tuning**
  Tuning character-based SMT on some development data.

Example: *dömmes (historical) – döms (modern)*

aligned bigrams:

```
dö öm mm me es s_
 |  |  \/   | /
dö öm  ms   s_
```

mapped to character alignments:

```
d ö m m e s
| | |/  |/
d ö m   s
```

Figure 3: Character-bigram alignment between a Swedish word in its historical and modern spelling respectively.

There are various settings and approaches that can be tested for each of these steps. In our experiments we will look at some important aspects of performing these tasks.

Interestingly, it is possible to adapt well-known alignment techniques to a lower level when moving from word-based to character-based models as we will see in our experiments. Word alignment can in fact be modeled in the same way as sentence alignment is modeled otherwise and character alignment can be performed successfully using word alignment techniques.

For word alignment, we can assume a strong correlation between the length of the original spelling and the modern spelling of the same word. Futhermore, we can assume that corresponding words contain many corresponding characters as well and that there should be no re-ordering of words in the normalised version, which leads to monotonic alignments. This is exactly what is used in common length-based and lexical matching-based sentence alignment algorithms. We therefore apply a standard sentence alignment tool for the word alignment task, *hunalign* (Varga et al., 2005), that combines both features; length correlation and lexical matches.

Character alignment is a bit more tricky and the relation to word alignment is less clear. In text normalisation, we would definitely not expect a lot of distortion (which is an important part of common word alignment models) and the vocabulary size is not comparable to that of a word alignment task as pointed out by Nakov and Tiedemann (2012). We therefore include the extension to bigrams as suggested by the same authors and also compare the results with transducer-based alignments that are developed for character-level transformations.

Character-bigrams include a bit more contextual information that may help the alignment, which otherwise mainly uses context-independent parameters. Fortunately, links between bigrams can still easily be mapped back to single character alignments which we need for training our translation models. See Figure 3 for an illustration of this process.

Similar to standard phrase-based SMT, we then extract mappings between character sequences of up to seven characters which are consistent with the alignment between characters. The standard scoring techniques are applied as in word-based models.

Language modeling can be done in the same way as for word-based models but trained on character sequences. We use 10-grams as suggested by Nakov and Tiedemann (2012) and standard smoothing techniques.

Parameter tuning is the final task that needs to be performed for building the character-based SMT model. We experiment with a standard Minimum Error Rate Training (MERT) optimized towards BLEU over character sequences on the translation of single words. We also tried sentence-based decoding with word-level BLEU but the normalisation accuracy was lower than with our word-based decoding.

4.1 Evaluation

The performance of our SMT-based approach to tagging and parsing of historical text is evaluated in three aspects: 1) normalisation accuracy, i.e. the percentage of tokens in the automatically normalised version of the text that are identical to the manually modernised gold standard version, 2) tagging performance before and after normalisation, and 3) parsing performance before and after normalisation.

We will look at different alignment settings and data sizes when evaluating the normalisaton accuracy. Furthermore, we will compare the normalisation accuracy to two baselines, hereafter referred to as *unnormalised* and *baseline*:

1. **unnormalised**
   The proportion of tokens in the unnormalised source text with a spelling identical to the modern spelling.
2. **baseline**
   The normalisation accuracy achieved when simply replacing each historical word type with its most frequent modern version observed in the word-aligned training corpus (leaving previously unseen tokens unchanged).

First, we will look at the case of historical Icelandic texts before moving on to a case study on Swedish texts. Since the access to corpora and NLP tools differs between the languages, the experiments are not entirely the same for the two languages, and the results are thus not fully comparable between the languages. However, for both languages we show that the proposed approach is successful for automatic annotation of historical text.

5 Case Study 1: Icelandic

For training and tuning of the Icelandic character-based SMT system, we make use of Snorri Sturluson's *Edda* (the Uppsala version DG11), an Icelandic saga available both in its 14th century spelling and in a manually modernised version (Palsson, 2012). These versions were automatically aligned using hunalign, and the extracted alignments were manually corrected, resulting in a parallel corpus of 33,888 entries in total. Training and tuning sets were created by extracting every 10th sentence to the tuning set, and the rest of the sentences to the training set. This extraction resulted in a training set of 30,451 token pairs and a tuning set of 3,437 token pairs.

As a language model for the SMT system, we use a subset of the Tagged Icelandic Corpus of contemporary Icelandic texts (Helgadóttir et al., 2012). In total, this subset consists of 21,613,551 tokens distributed over 12 genres, including for example newspaper text, blog text,

parliamentary speeches and university essays. We would of course prefer to use the whole corpus, but at the time of writing, the full corpus has not yet been made available.

Evaluation of normalisation accuracy, as well as tagging results, is performed on a subset of *Ectors saga* from the 15th century (Loth, 1962). This text contains 20,811 tokens and is part of the Icelandic Parsed Historical Corpus, IcePaHC, a manually tagged and parsed diachronic corpus of texts spanning from 1150 to 2008 (Rögnvaldsson et al., 2012). In the IcePaHC corpus, the old texts have been manually modernised with regard to spelling. We also have access to the saga in its 15th century spelling, needed for evaluation purposes.

For tagging, we use the IceNLP tagger (Loftsson and Rögnvaldsson, 2007) trained on the Icelandic Frequency Dictionary corpus (IFD) of approximately 500,000 tokens from the time period 1980–1990 (Pind, 1991).

Evaluation of parsing performance was not possible, since we do not have access to a parsed gold standard corpus in a suitable format. The IcePaHC corpus does include syntactic information, but in an annotation scheme that is not easily mapped to the one produced by the IceNLP parser.

5.1 Word Alignment

To train a character-based SMT system on the *Edda*, the historical and the modern version of the text has to be aligned on a word level. For this purpose we use hunalign (Varga et al., 2005), which is in fact a sentence aligner rather than a word aligner. As stated in section 4, for our specific alignment task we can however assume monotonic alignments without re-ordering and with a strong correlation between the length of the original spelling and the modern spelling, which would be a suitable task for sentence alignment. We tried four different ways of using hunalign for our word alignment task:

1. **no split**
   Input data is one token on each line, with empty lines denoting sentence boundaries.
2. **no split +realign**
   Same as ”no split”, but with the additional flag *-realign*, in which the aligner is run in three phases, heuristically building a dictionary based on the identified sentence pairs (in our case word pairs).
3. **split**
   Same as ”no split”, but with whitespace separating all characters.
4. **split +realign**
   Same as ”split”, but with the additional *-realign* flag, as described above.

Automatic alignment will inevitably introduce noise in the training data. Since the modern version is stated to be a modernisation of spelling, not including syntactic normalisation, the word alignment task is however easier than in an ordinary translation setting. To evaluate the impact of the different alignment methods on the end result, we ran experiments based on the GIZA++ unigram setting for character alignment, with training data automatically generated from the four alignment methods described above, as well as with training data that had been manually corrected. As illustrated in Table 1, splitting the words into their separate characters has a positive effect on the alignment results. With the split +realign setting, normalisation accuracy is 79.2% as compared to 78.8% for the no split setting.

Approximately two thirds (64.8%) of the original tokens already had a spelling that was identical to the modern spelling. With the simplistic baseline, replacing each historical word type with its most frequent modern version observed in the word-aligned training data, normalisation accuracy increases to 69.8%, which is far from the results achieved for the SMT model.

It is also worth mentioning, that the normalisation accuracy for automatically generated training data is close to the accuracy achieved for manually corrected training data; 79.2% in the best hunalign setting as compared to 83.9% for manually corrected data. Hence, if no word-aligned historical-modern data is available, automatic alignment techniques may successfully be used to automatically create such a parallel corpus.

Hunalign Model	**Accuracy**
unnormalised	64.8%
baseline	69.8%
no split	78.8%
no split +realign	78.8%
split	79.1%
split +realign	79.2%
manual alignment	83.9%

Table 1: Normalisation accuracy with different methods for alignment of the training data. Unnormalised = Percentage of tokens in the unnormalised text with a spelling identical to the modern spelling. Baseline = Normalisation accuracy when replacing each historical word type with its most frequent modern version observed in the word-aligned training corpus.

5.2 Character Alignment

As mentioned in Section 3, we have experimented on using GIZA++ and m2m aligner models with different settings for training the character-based SMT systems. The results for different settings on the Icelandic training corpus are summarised in Table 2.

Alignment	**Accuracy**
unnormalised	64.8%
baseline	69.8%
giza unigram	83.9%
giza bigram	83.5%
m2m 1:1	81.0%
m2m 2:2	83.6%

Table 2: Normalisation accuracy for different character alignment models. Unnormalised = Percentage of tokens with a spelling identical to the modern spelling before normalisation. Baseline = Normalisation accuracy when replacing each historical word type with its most frequent modern version observed in the word-aligned training corpus. Giza uses standard word alignment models (for character unigrams and bigrams) and m2m uses the WFST with single character edit operations (1:1) and multi-character operations (2:2).

As previously stated, approximately two thirds (64.8%) of the original tokens already had a spelling that was identical to the modern spelling. One could have expected this figure to be higher, since modern Icelandic is generally seen as close in spelling to its historical variants. However, in addition to spelling variation, Icelandic has a rich morphology that has changed over time, which is also influencing the words in terms of string similarity. The SMT models were able to successfully capture some of these differences, resulting in 83.9% correctly normalised tokens in the best setting, i.e. the GIZA++ unigram setting. Using more informative bigrams instead of unigrams surprisingly lead to a drop in normalisation accuracy from 83.9% to 83.5%. We also tried the m2m aligner, with single character edit operations (1:1) and multi-character operations (2:2), which was slightly less successful for our normalisation task as compared to the GIZA++ unigram model.

5.3 Training Data

In Section 5.2, we have shown that training a character-based SMT system on a parallel corpus of historical and modern spelling is successful for normalising historical text to a modern spelling. So far, we have used a parallel corpus of 33,888 token pairs for training and tuning. It might be the case that only a smaller data set is available in this form, or that such a parallel corpus is not available at all and would need to be manually created. To see whether an even smaller corpus would be sufficient for achieving reasonable results, we experimented on different sizes of the parallel corpus used for training. As expected, the more training data, the better normalisation results (in general), as presented in Figure 4. However, with only 1,000 token pairs, we already achieve 76.5% normalisation accuracy, as compared to 83.9% for the entire corpus.

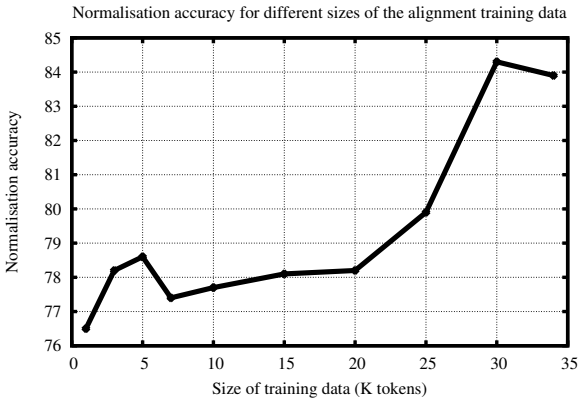

Figure 4: Normalisation accuracy when varying the size of the alignment training data.

5.4 Language Modeling

For the language model, we have experimented on varying the size and genres included in the training data. A motivation for this is that we would like our normalisation approach to be useful for any language where the modern version of the language has a Basic Language Resource Kit, BLARK (as defined in Krauwer et al. (2004)). Such a BLARK may contain corpora of varying sizes, sometimes including several genres and sometimes including for example newspaper text only. We have performed experiments on varying the size of the language model training data for Icelandic, from 1 million tokens as a minimum to including all 21,613,551 tokens of the corpus. Furthermore, we have experimented on using newspaper text only, as compared to including all the genres of the corpus. The results are presented in Figure 5.

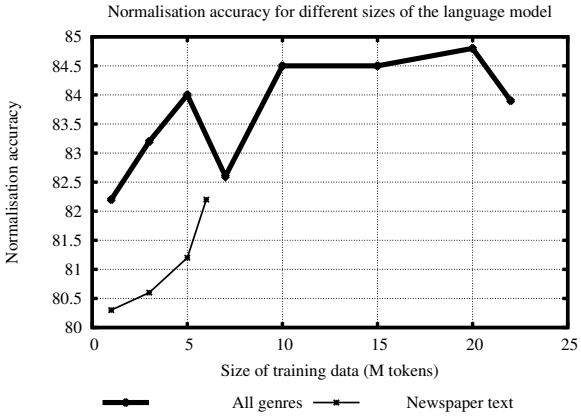

Figure 5: Normalisation accuracy, when varying the size and text types included in the language model.

As seen from the results, including a corpus of different genres in the language model shows slightly better results than using newspaper text only. However, there is not a huge difference between the two types of corpora. For 5 million words of newspaper text, a normalisation accuracy of 81.2% is achieved, as compared to 84.0% for the sampled corpus. It is also worth mentioning that whereas the newspaper corpus shows the expected increase in normalisation accuracy when more data is added, this relation is not as clear-cut for the sampled corpus where the addition of more data in some cases leads to a drop in normalisation accuracy. This could be due to larger variation in the sampled training data, meaning that adding new data sometimes distort the language model.

Also note that including only 1 million (sampled) tokens results in a normalisation accuracy of 82.2%, which is already close to the 83.9% achieved when including the full corpus in the language model.

5.5 Tagging

Without normalisation, the IceNLP tagger is able to correctly annotate 46.7% of the tokens in the test text (*Ectors saga*). Besides the spelling variation, one reason for the rather low tagging

accuracy could be that the manually tagged corpus is, as stated earlier, based on modernised spelling. The modern spelling also seem to include morphological changes, meaning that the gold standard tag is in fact not in all cases the correct tag for the token when its historical spelling is preserved. Despite this deficiency in the gold standard corpus, tagging accuracy increases by 10 percentage units after normalisation, to 56.6%, see further Table 3.

	Accuracy
unnormalised	46.7%
baseline	49.9%
normalised	56.6%

Table 3: Accuracy for Icelandic part-of-speech tagging. Baseline = Normalisation by replacing each historical word type with its most frequent modern version observed in the word-aligned training corpus. Normalised = Normalisation by the Giza unigram approach.

6 Case Study 2: Swedish

For training, tuning and evaluation in the Swedish setting, we use a corpus of court records and church documents from the period 1527–1812. These texts are available in both their original and their modernised spelling (see further Pettersson et al. (2012) for more details on the corpus). For training we extracted 28,237 token pairs, whereas the tuning set consists of 2,590 (non-overlapping) token pairs and the test set includes 33,544 (non-overlapping) token pairs, equally distributed over the texts in the corpus. The test set is the same subset of the corpus as is used in Pettersson et al. (2012).

As a language model, we make use of the Stockholm-Umeå Corpus, SUC, a balanced corpus consisting of approximately one million tokens extracted from a number of different text types representative of the Swedish language in the 1990s (Ejerhed and Källgren, 1997).

For tagging, we use HunPOS (Halácsy et al., 2007), a free and open source reimplementation of the HMM-based TnT-tagger by Brants (2000). In our experiments, we use HunPOS with a Swedish model based on the SUC corpus.

For parsing, we use MaltParser version 1.6, a data-driven dependency parser developed by Nivre et al. (2006a). In our experiments, the parser is run with a pre-trained model[1] for parsing contemporary Swedish text, based on the Talbanken section of the Swedish Treebank (Nivre et al., 2006b).

6.1 Character Alignment

We have experimented on using GIZA++ and m2m aligner models with different settings for training the character-based Swedish SMT systems, as presented in Table 4. Similar to the results achieved for Icelandic, the best-performing setting is to train a GIZA++ unigram model, which outperforms the more informative bigram model as well as the m2m aligner models. In the GIZA++ unigram setting, the percentage of tokens with a spelling identical to the gold standard spelling increases from 64.6% for the original spelling, to 92.3% after normalisation. It is interesting to note that the baseline method of simply replacing each historical word type with its most frequent modern version observed in the word-aligned training corpus also

[1] http://maltparser.org/mco/swedish_parser/swemalt.html

has a large positive effect on normalisation accuracy, increasing the number of tokens with a modern spelling from 64.6% to 86.1%. For Icelandic this effect was considerably less significant, increasing the number of tokens with a modern spelling from 64.8% to 69.8%. One reason for the larger impact on Swedish might be that training and test sets are extracted from the same historical corpus (though non-overlapping), whereas for Icelandic the training set is extracted from one text and the test set from another text.

Alignment	**Accuracy**
unnormalised	64.6%
baseline	86.1%
giza unigram	92.3%
giza bigram	91.8%
m2m 1:1	91.4%
m2m 2:2	89.7%

Table 4: Normalisation accuracy for different character alignment models. Unnormalised = Percentage of tokens with a spelling identical to the modern spelling before normalisation. Baseline = Normalisation accuracy when replacing each historical word type with its most frequent modern version observed in the word-aligned training corpus. Giza uses standard word alignment models (for character unigrams and bigrams), and m2m uses the WFST with single character edit operations (1:1) and multi-character operations (2:2).

6.2 Tagging and Parsing

For the Swedish setting, we do not have access to a linguistically annotated gold standard for evaluating tagging and parsing performance. However, in the test corpus described in Section 6, all the verbs and their complements have been manually annotated as such. Therefore, we decided to indirectly evaluate the performance of a tagger based on precision and recall for verb identification before and after normalisation of the input text. Similarly, parsing performance is indirectly evaluated based on the proportion of correctly detected verb complements.

Tagging The results presented in Table 5 show that normalisation has a large positive impact on precision and recall measures for verb identification. Without normalisation, an F-score of 70.4% is achieved for this task, as compared to 83.5% for the baseline approach to normalisation and 88.7% for the SMT approach to normalisation. The largest increase is in recall, where 89.8% of the verbs are identified after normalisation, as compared to 64.2% for the original spelling. Precision also increases substantially, from 77.9% to 87.7%.

Parsing MaltParser produces dependency trees labeled with grammatical functions, which can be used to identify different types of complements. In this setup, we define *exact matches* as cases where the detected verb complement is identical to the corresponding gold standard complement. Likewise, we define *partial matches* as cases where the detected verb complement has a non-empty overlap with the corresponding gold standard complement. Consider for example the gold standard analysis *effterfrågat [om sinss manss dödh]* (asked [about her husband's death]). A system output where the head noun of the complement is missing, will still be regarded as partially correct: *effterfrågat [om sinss manss]* (asked [about her husband's]).

	Precision	Recall	F-score
unnormalised	77.9%	64.2%	70.4%
baseline	84.0%	83.0%	83.5%
normalised	87.7%	89.8%	88.7%

Table 5: Precision and recall measures for verb identification based on tagging. Baseline = Normalisation by replacing each historical word type with its most frequent modern version observed in the word-aligned training corpus. Normalised = Normalisation by the Giza unigram approach.

As shown in Table 6, the proportion of correctly identified verb complements (exact and partial matches) increases from 32.9% for the original historical spelling to 42.6% after baseline normalisation, and 46.2% after SMT normalisation.

	Exact Match	Partial Match	Exact or Partial Match
unnormalised	23.8%	9.1%	32.9%
baseline	30.2%	12.4%	42.6%
normalised	33.2%	13.0%	46.2%

Table 6: Precision and recall measures for verb complement detection based on parsing. Baseline = Normalisation by replacing each historical word type with its most frequent modern version observed in the word-aligned training corpus. Normalised = Normalisation by the Giza unigram approach.

7 Previous Studies

There are several approaches to spelling modernisation of historical text that have been described in previous studies. One of the earlier examples is the VARD tool (VARiant Detector), which is based on dictionary lookup, mapping 16th to 19th century English spelling to modern spelling. Evaluation was performed on a set of 17th century texts, and compared to the performance of modern spell checkers. Between a third and a half of all tokens were correctly normalised by both VARD and MS Word, whereas approximately one third of the tokens were correctly normalised only when using VARD. The comparison between VARD and Aspell showed similar results (Rayson et al., 2005).

Pettersson et al. (2012) tried a rule-based letter transformation approach, where a set of 29 hand-crafted normalisation rules was produced on the basis of a 17th century court records text. The resulting rule set was applied to a gold standard corpus of 33,544 tokens from the period 1527–1812, i.e. the same test corpus as is used in our Swedish case study. The normalisation had a positive effect on texts from all centuries covered in the corpus. On average, approximately 73% of the tokens were correctly normalised using this method.

A data-driven approach based on Levenshtein similarity was presented by (Bollmann et al., 2011) for normalisation of Early New High German. Normalisation rules were automatically derived by means of the Levenshtein edit distance, based on a word-aligned parallel corpus consisting of the Martin Luther bible in its 1545 edition and its 1892 version, respectively. Using this normalisation technique, the proportion of words with a spelling identical to the modern spelling increased from 65% in the original text to 91% in the normalised text.

8 Conclusion

In this paper, we have shown that using character-based SMT techniques for normalising historical text to a modern spelling, is successful when applying modern taggers and parsers to analyse historical text. In the Swedish case study, the proportion of tokens in the historical text that are identical to the modern spelling increases from 64.6% to 92.3% in the best setting. This results in an increase in F-score for verb identification (based on tagging) from 70.4% before normalisation to 88.7% after normalisation. Accordingly, the proportion of correctly identified verb complements (based on parsing) increases to the same extent.

Furthermore, we have shown that it is possible to achieve good results without (or with little) manual efforts, since only a small amount of training data is needed to achieve reasonable results. With only 1,000 tokens available in a historical and a modern spelling, a normalisation accuracy of 76.5% is achieved for Icelandic, as compared to 83.9% in the best setting. If no word-aligned training data at all is available, automatic sentence alignment methods may successfully be used for automatically creating training data, as shown in section 5.1.

The data-driven nature of our approach makes it language-independent, and we believe that our method is generally applicable to languages for which there is a corpus of modern text available, as well as a small data set of historical texts with both historical and modern spelling. So far, we have evaluated our approach for Swedish and Icelandic. In the future, we would like to extend our experiments to texts from other languages, time periods and genres. It would also be interesting to explore ways of normalising not only spelling, but also morphological and syntactic differences in the historical texts.

References

Bollmann, M., Petran, F., and Dipper, S. (2011). Rule-based normalization of historical texts. In *Proceedings of the Workshop on Language Technologies for Digital Humanities and Cultural Heritage*, pages 34–42, Hissar, Bulgaria.

Brants, T. (2000). TnT - a statistical part-of-speech tagger. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP)*, Seattle, Washington, USA.

Ejerhed, E. and Källgren, G. (1997). Stockholm Umeå Corpus. Version 1.0. Produced by Department of Linguistics, Umeå University and Department of Linguistics, Stockholm University. ISBN 91-7191-348-3.

Halácsy, P., Kornai, A., and Oravecz, C. (2007). HunPos - an open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 209–212, Prague, Czech Republic.

Helgadóttir, S., Svavarsdóttir, A., Rögnvaldsson, E., Bjarnadóttir, K., and Loftsson, H. (2012). The tagged icelandic corpus (mím). In *Proceedings of the Workshop on Language Technology for Normalisation of Less-Resourced Languages*, pages 67–72.

Jiampojamarn, S., Kondrak, G., and Sherif, T. (2007). Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2007)*, pages 372–379, Rochester, NY.

Krauwer, S., Maegaard, B., Khalid, C., and Damsgaard Jørgensen, L. (2004). Report on Basic Language Resource Kit (BLARK) for Arabic.

Loftsson, H. and Rögnvaldsson, E. (2007). IceNLP: A natural language processing toolkit for Icelandic. In *Proceedings of InterSpeech, Special session: Speech and language technology for less-resourced languages*, Antwerp, Belgium.

Loth, A., editor (1962). *Late Medieval Icelandic Romances I*. Kaupmannahöfn, Copenhagen.

Matthews, D. (2007). Machine transliteration of proper names. Master's thesis, School of Informatics.

Nakov, P. and Tiedemann, J. (2012). Combining word-level and character-level models for machine translation between closely-related languages. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 301–305, Jeju Island, Korea. Association for Computational Linguistics.

Nivre, J., Hall, J., and Nilsson, J. (2006a). MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of the 5th international conference on Language Resources and Evaluation (LREC)*, pages 2216–2219, Genoa, Italy.

Nivre, J., Nilsson, J., and Hall, J. (2006b). Talbanken05: A Swedish treebank with phrase structure and dependency annotation. In *Proceedings of the 5th international conference on Language Resources and Evaluation (LREC)*, pages 24–26, Genoa, Italy.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of ACL'03*, pages 160–167, Sapporo, Japan.

Palsson, H., editor (2012). *The Uppsala Edda*. Viking Society for Northern Research.

Pettersson, E., Megyesi, B., and Nivre, J. (2012). Rule-based normalisation of historical text - a diachronic study. In *Proceedings of the First International Workshop on Language Technology for Historical Text(s)*, Vienna, Austria.

Pind, J., editor (1991). *Icelandic Frequency Dictionary*. Institute of Lexicography, Reykjavik, Iceland.

Rayson, P., Archer, D., and Nicholas, S. (2005). VARD versus Word – A comparison of the UCREL variant detector and modern spell checkers on English historical corpora. In *Proceedings from the Corpus Linguistics Conference Series on-line e-journal*, volume 1, Birmingham, UK.

Rögnvaldsson, E., Ingason, A. K., sson, E. F. S., and Wallenberg, J. (2012). The icelandic parsed historical corpus (icepahc). In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Varga, D., Németh, L., Halácsy, P., Kornai, A., Trón, V., and Nagy, V. (2005). Parallel corpora for medium density languages. In *Proceedings of the RANLP*, pages 590–596.

Vilar, D., Peter, J.-T., and Hermann, N. (2007). Can we translate letters? In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 33–39, Prague, Czech Republic. Association for Computational Linguistics.