Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 3354–3366
Marseille, 20-25 June 2022

Automatic Normalisation of Early Modern French

Rachel Bawden[1] **Jonathan Poinhos**[2] **Eleni Kogkitsidou**[2]
Philippe Gambette[2] **Benoît Sagot**[1] **Simon Gabay**[3]
[1]Inria, Paris, France
[2]LIGM (UMR 8049), Université Gustave Eiffel, CNRS, 77454 Marne-la-Vallée, France
[3]Université de Genève, Switzerland
firstname.lastname@{inria.fr,univ-eiffel.fr,unige.ch},

Abstract

Spelling normalisation is a useful step in the study and analysis of historical language texts, whether it is manual analysis by experts or automatic analysis using downstream natural language processing (NLP) tools. Not only does it help to homogenise the variable spelling that often exists in historical texts, but it also facilitates the use of off-the-shelf contemporary NLP tools, if contemporary spelling conventions are used for normalisation. We present FREEM$_{norm}$, a new benchmark for the normalisation of Early Modern French (from the 17th century) into contemporary French and provide a thorough comparison of three different normalisation methods: ABA, an alignment-based approach and MT-approaches, (both statistical and neural), including extensive parameter searching, which is often missing in the normalisation literature.

Keywords: Digital Humanities, Normalisation, Spelling, Modern French, Machine Translation, Historical

1. Introduction

Computational approaches have recently been playing an increasing role in the humanities (Gabay, 2021), especially concerning the study of textual documents. Historical documents are particularly interesting, as they are an invaluable source of historical information and are crucial witnesses of language evolution.

Whether documents are to be studied manually by philologists and literary experts or analysed automatically using downstream natural language processing (NLP) tasks such as part-of-speech (PoS) tagging and parsing, a useful preliminary step is *normalisation*, which consists in modernising the spelling of the documents to conform to contemporary spelling conventions. Normalisation has the effect of (i) reducing spelling variation present in historical documents, often written at a time spelling was not standardised, and (ii) reducing the gap between the historical state of the language and the contemporary state. Importantly, this allows us to apply off-the-shelf NLP tools to old texts and limit the performance drop that can usually be expected, for example for tagging and parsing (Pettersson et al., 2013b) or geographical named entity recognition (Kogkitsidou and Gambette, 2020).

There has been a considerable amount of previous research in historical spelling normalisation, with a range of methods being developed, including manually developed rules (Porta et al., 2013; Baron and Rayson, 2009; Riguet, 2019), those exploiting edit distances and other external resources such as lexicons (Mitankin et al., 2014) and machine translation (MT) approaches, both statistical (Scherrer and Erjavec, 2013; Domingo and Casacuberta, 2018a) and neural (Bollmann and Søgaard, 2016; Hämäläinen et al., 2018). Despite this, questions still remain regarding which method is the most effective, particularly between statistical MT (SMT) and neural MT (NMT) approaches. There has for example been little research in optimising these models for the particular task, which could lead to false conclusions being drawn about which model is best; as has been previously shown for low-resource tasks, neural models in particular are sensitive to model size, training parameters and the degree of subword segmentation applied to texts (Sennrich and Zhang, 2019; Fourrier et al., 2021).

Our focus in this paper is on the normalisation into contemporary French of Early Modern French (also known as Modern French or Classical French), which is French from the 17th century. Despite several recent efforts (Gabay and Barrault, 2020; Gabay et al., 2019; Riguet, 2019), there has so far been very little research carried out on spelling normalisation for historical French, and so we aim to fill this gap. Figure 1 illustrates a few of the normalisation types observed, from simple typographic changes (e.g. *ſ* → *s*), changes to segmentation (*long temps* 'a long time' → *longtemps*), changes reflecting language change (*eſtoit* '(s/he) was' → *était*) and the use of classical false etymological spellings (e.g. *ç* being used in Modern French *ſçavoir* 'to know' as a link to Latin *scire*, from which it does not originate).

In this paper, we present the parallel normalisation corpus, FREEM$_{norm}$ (for Early Modern French), on which we train and evaluate, and, in addition to baseline models, we compare three methods: (i) an alignment-based approach, called ABA, using automatically learned word correspondences from a parallel corpus, (ii) phrase-based SMT, and (iii) NMT, comparing an LSTM model (Bahdanau et al., 2015) and a Transformer (Vaswani et al., 2017). We find that despite extensive parameter optimisation for NMT models, SMT produces the best results overall, with all methods largely exceeding the baselines. Our comparison shows that the methods exhibit quite different be-

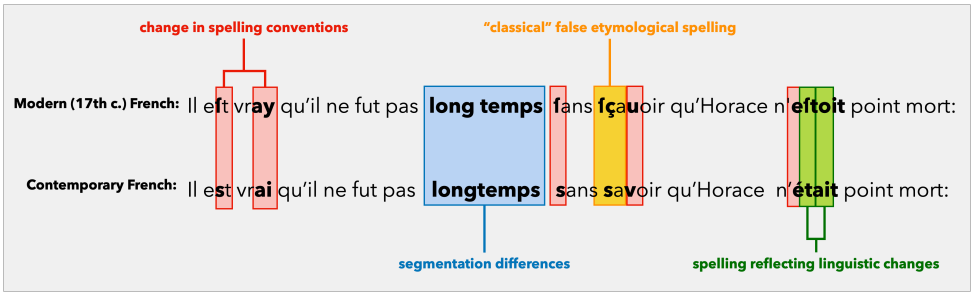

Figure 1: A Modern French sentence and its contemporary French normalisation.

haviour in terms of how conservative or inventive they are, which could be useful information depending on the downstream task (e.g. as a pre-annotation tool for manual annotation or a downstream NLP application). Our main contributions can be summarised as follows:

- Introduction of a new benchmark for the normalisation of Modern French, which can be used in further research.
- Extensive experiments comparing an alignment-based approach (ABA) with three MT approaches (SMT, LSTM and Transformer), with best results achieved by SMT. We also show that a lexicon-based post-processing step can systematically improve over all other methods tested.
- We freely distribute the data,[1] scripts and state-of-the-art normalisation models.[2,3]

2. Related Work

A considerable amount of work has been carried out in historical spelling normalisation, across various languages, with research dating back to the 1980s (Fix, 1980). A range of different approaches have been developed, including rule-based (Porta et al., 2013; Riguet, 2019), the use of various types of edit-distance (Hauser and Schulz, 2007; Bollmann, 2012; Pettersson et al., 2013a) and MT-style approaches, both statistical (Vilar et al., 2007; Scherrer and Erjavec, 2013; Ljubesic et al., 2016; Domingo et al., 2017) and neural (Korchagina, 2017; Domingo and Casacuberta, 2018b; Tang et al., 2018). Interestingly, all of these approaches remain useful today, thanks to their different strengths, depending on the type of normalisation and the amount of data available (Bollmann, 2019).

2.1. Word Lists, Rules and Edit-based Methods

Approaches relying on word lists, consisting in simply replacing historical variants by their normalised equivalent have been developed in several languages: English (Reynaert et al., 2012), German, Portuguese (Piotrowski, 2012) and Slovene (Erjavec et al., 2011).

Many rule-based and edit-distance-based approaches are unsupervised (i.e. they do not require parallel data), which is a considerable advantage, especially for historical varieties for which annotated data is not readily available. Rules can be developed manually by experts (Porta et al., 2013; Baron and Rayson, 2009; Riguet, 2019) or be extracted from a comparison of historical and modern word lists or parallel data if this is available (Bollmann et al., 2011).

The use of edit distance, using for example Levenshtein distance is often a strong baseline (Pettersson et al., 2013a), due to the fact that the surface forms of historical and contemporary spellings are often very similar and the alignment between both words and characters in the two varieties is almost perfectly monotonic. Basic edit-distance can be enhanced with specific weights for different edits (Bollmann et al., 2011) or based on characters or character groups (Hauser and Schulz, 2007; Bollmann, 2012), given the observation that certain errors are more serious than others.

2.2. Normalisation as MT

MT approaches to the problem have been popular, with the historical and modern states of the language being treated as the source and target languages respectively.

Characters, Subword or Words? Most previous research has focused on character-based MT, which models transformations at the level of individual characters (Vilar et al., 2007; Scherrer and Erjavec, 2013; Pettersson et al., 2013b; Domingo and Casacuberta, 2021), which makes sense for the task of spelling normalisation, as it often involves local transformations and largely monotonic alignments between source and target sentences. However, there has since been work exploring word translation, subword translation (Tang et al., 2018) or a mixture of these (Vilar et al., 2007; Domingo and Casacuberta, 2021). It is rare however for works in historical spelling normalisation to explore the optimal degree of segmentation, although Tang et al. (2018) do find subwords to be more effective than character-based: character-based segmentation offers a greater possibility for generalisation with the caveat that it requires the model to learn to translate longer sequences and learn patterns better, whereas word or subword segmentation can exploit models' ability to

[1] https://doi.org/10.5281/zenodo.5865428
[2] https://github.com/rbawden/ModFr-Norm
[3] See https://freem-corpora.github.io for the project page.

memorise, but may run the risk of limited generalisation, especially to unseen or less frequent words.

SMT or NMT? The first approaches were with SMT (Koehn et al., 2007), which proved more effective than rule-based and edit-distance based approaches (Pettersson et al., 2014; Hämäläinen et al., 2018; Bollmann, 2019), when there is parallel data available, and even when this data is produced synthetically (Scherrer and Erjavec, 2013; Domingo and Casacuberta, 2018a). NMT approaches to historical spelling normalisation were developed as it took off in the domain of general MT (Bollmann and Søgaard, 2016; Hämäläinen et al., 2018). Comparisons between SMT and NMT show different results, with SMT being superior in some cases (Domingo and Casacuberta, 2018a), and NMT in others (Bollmann, 2019), provided enough parallel data is available (Bollmann, 2019). Importantly, the methods appear to have different behaviours and therefore their own strengths and weaknesses, meaning that a single method (including rule-based approaches) is not necessarily a systematically better choice (Hämäläinen et al., 2018; Robertson and Goldwater, 2018).

Word Translation vs. Sentence Translation A considerable portion of the research in historical normalisation is based on the normalisation of word lists, so of words in isolation. However, as discussed in (Ljubesic et al., 2016), it can be beneficial in some contexts to normalise whole sentences (where there is ambiguity in the normalised form that should be chosen). This has the disadvantage of creating longer sequences to process, but is necessary in order to hope to handle all phenomena. The development of parallel corpora rather than word lists has encouraged research in this direction (Tjong Kim Sang et al., 2017; Gabay and Barrault, 2020; Ortiz Suarez et al., 2022).

2.3. Normalisation for Historical French

Despite there being a plethora of research on historical spelling normalisation, little research has been done so far on historical French, with most work focusing on Dutch, German, Hungarian, Slovene, and Swedish, helped by the existence of benchmark data (Dipper and Schultz-Balluff, 2013) and shared tasks (Ljubesic et al., 2016; Tjong Kim Sang et al., 2017).

A collaborative word list associating normalised versions of historical words in French was started in 2009 on the Wikisource digital library,[4] which is available for automatic normalisation through word substitution (The French Wikisource Community, 2022). Recently, there has been some preliminary research, with the development of a parallel corpus for the normalisation of Modern French (from the 17th c.) (Ortiz Suarez et al., 2022) and first baselines, including rule-based (Riguet, 2019) and NMT-style approaches (Gabay et al., 2019; Gabay and Barrault, 2020). Gabay and Barrault (2020) compare character-based SMT and NMT at different granularities (words, subwords and characters): NMT outperformed SMT, and for NMT, the best input representations were found to be words, then characters, then subwords. However, they do not seem to perform a comparison of different levels of subword segmentation or of different sizes of architecture, which has been shown to be important when drawing conclusions about the usability of NMT in low-resource settings (Sennrich and Zhang, 2019).

[4]On 17th January 2022, the whole list contains 15,470 words and expressions and their normalised equivalents.

3. Approaches Compared

We present and compare several approaches, representing a wide range of techniques: (i) an alignment-based method using a parallel corpus (Section 3.1), (ii) statistical MT (Section 3.2.1), (iii) neural MT, testing both LSTM and Transformer models (Section 3.2.2). In addition to comparing these approaches to two baselines described in Section 6.1, we also assess the impact of a lexicon-based post-processing described in Section 3.3.

3.1. ABA: Alignment-based

The ABA method (short for alignment-based method), is a hybrid approach consisting of (i) word-level transformation rules that are automatically learned from an aligned corpus and (ii) character-level transformation rules, which were manually designed by observing frequent character transformations in the aligned corpus. The ABA normalisation method, which has similarities with the approach of VARD2 developed for English (Baron and Rayson, 2009), works as follows.

Creation of a Word Substitution Lexicon The first step is to learn a word replacement lexicon using a parallel training set. This is done using the classical dynamic programming Needleman-Wunsch alignment algorithm (Needleman and Wunsch, 1970) to optimally align tokenised parallel sentences at the token level, adding a score of 4 for matching words in lowercase (or for *et* and *&* 'and' which are considered equivalent) and a penalty of -1 for word insertions, deletions or mismatches if the non-matching words have a weighted Levenshtein distance of at least 4 or at least the length of each word. For mismatches between words at weighted Levenshtein distance $d < 4$ and strictly smaller than the length of both words, $4 - d$ is the mismatch score taken into account by the alignment algorithm. Note that the weighted Levenshtein distance is computed with a penalty of 1 for insertions and deletions and 2 for character mismatches. These scores were adjusted experimentally after considering the alignment results on a training corpus.

Substitution Step The second step uses this replacement lexicon as well as a contemporary French lexicon built by combining `Morphalou 3.1` (ATILF, 2019) with lexicons of proper nouns developed for CasEN 1.4 (Maurel et al., 2011): `CasEN_Dico.dic`, `Prolex-Unitex-BestOf_2_2_fra.dic` (CasEN Team, 2019) and `Prolex-Unitex_1_2.dic` (Prolex Team, 2013). It proceeds in the following way:

after simple tokenisation[5] of the input text, for each token: 1) if it is present in the contemporary French lexicon, it is kept as it is; 2) otherwise, if it is present in the word replacement lexicon, it is replaced by the associated normalised version in this lexicon; 3) otherwise, it is transformed by a combination of character replacement rules detailed in Appendix A, designed after careful analysis of the aligned words in the training corpus and available in the `apply_rules` function of the `modern.py` script in ABA's distribution:[6] among the obtained candidate tokens, the first one found in the contemporary French lexicon is selected; 4) otherwise, if no candidate generated by character transformation rules is selected, then the original token is kept.

3.2. MT: SMT and NMT

Following promising results for other languages (Scherrer and Erjavec, 2013; Tang et al., 2018) and Modern French (Gabay et al., 2019; Gabay and Barrault, 2020), we provide a comparison of phrase-based statistical MT and NMT.

3.2.1. Phrase-based SMT

The aim of SMT is to automatically find the most probable translation $\hat{t}$ given a source sentence s such that $\hat{t} = \text{argmax}_{t \in T} P(s|t) P(t)$, where $P(s|t)$ models the adequacy of translation, and $P(t)$ the target language model probability, which can be seen as a measure of the fluency/grammaticality of the prediction. The state of the art in SMT is phrase-based MT, where a prediction's score is the sum of scores from various scoring components, including a phrase table (for the translation probability), a language model (for the language model probability), a reordering (or distortion) model and a length penalty. The main implementation used for phrase-based SMT is the Moses toolkit (Koehn et al., 2007), which we use here in this paper.

Phrase-based SMT was the state of the art in MT until around 2015, when NMT first outperformed it (Bahdanau et al., 2015). The main disadvantages of SMT with respect to NMT is the limited ability to model longer distance dependencies and to model semantic relationships between input units, given that probabilities are calculated based on discrete surface forms rather than continuous representations. It nevertheless remains relevant in certain settings, notably when little parallel training data is available (Trieu et al., 2017; Fourrier et al., 2021). For historical spelling normalisation, some works have shown that it can outperform neural approaches, particular in these lower-resource settings (Domingo and Casacuberta, 2018a).

3.2.2. NMT (LSTM and TRANSFORMER)

NMT uses neural networks to find the most probable translation. The standard architecture is an encoder-decoder with an attention mechanism (Bahdanau et al., 2015). The role of the encoder is to encode the source sequence and of the decoder to sequentially produce the target sequence, given the previously translated words and a representation of the input sequence specific to that decoding step (calculated using attention). Importantly, these models work with continuous representations of words, allowing for a greater capacity to generalise across forms and an improved handling of complex linguistic phenomena. The first such models were based on recurrent neural networks (using recurrent units such as LSTM for example), involving sequentially encoding the input and sequentially decoding the output. The current state of the art is the Transformer, which replaces recurrence with self-attention (Vaswani et al., 2017). Transformers have the advantage of speed in training and tend also to perform better, although this does not always hold for very low-resource settings (Fourrier et al., 2021).

NMT model performance is sensitive to the size of the architecture, subword segmentation and training parameters. Sennrich and Zhang (2019) show that previous conclusions about the superiority of SMT systems over NMT in low-resource scenarios do not necessarily hold as long as the NMT parameters are well chosen, highlighting the need to perform adequate parameter search before drawing conclusions. In line with this, we perform extensive hyper-parameter searches of both LSTM and Transformer models (Section 6.3).

3.3. Optional Le*fff*-based post-processing

All three approaches described above rely on parallel training data. Despite the generalisation capabilities of such models, it might be the case that rare situations are not properly dealt with. On the other hand, large-scale lexicons of contemporary French, such as the Le*fff* (Sagot, 2010), can provide high-coverage lexical information regarding the target language of the normalisation process.

Based on this observation, we developed a lexicon-based post-processing tool that can be used after any normalisation model and is based on the Le*fff* (version 3.4). It relies on the idea that a normalised text should mostly contain words known to a large-scale contemporary French lexicon. Any token (whitespace- and/or punctuation-separated character sequence) that does not begin with a capital letter (to avoid proper nouns) and that is unknown to the lexicon is eligible for further normalisation. For every such token, we compute a list of possible normalisations based on a small list of permitted transformations.[7] We then look up all normalisation candidates in our lexicon. If exactly one of the normalisation candidates is known to our lexicon, we replace the input token with this candidate. In all other cases, we leave the token unchanged.

[5] Splitting the sentence on whitespace, the characters . , ! ? ; : and both kinds of apostrophe.

[6] `https://github.com/johnseazer/aba`.

[7] The rules: $Vs \rightarrow \hat{V}$ where V is any vowel, *es* → é, add each possible diacritic to each non-diacritised letter that can have a diacritic (e.g. $u \rightarrow \ddot{u}$), v → u when preceded by a vowel, u → v when preceded by a consonant, $i \rightarrow$ y.

4. Evaluating Normalisation

In terms of automatic metrics, the most commonly used are translation edit rate (TER), word accuracy (based on the gold normalised tokens, non-symmetrised) and some works have used traditional metrics for MT (Gabay and Barrault, 2020), in particular BLEU (Papineni et al., 2002) and CHRF (Popović, 2015). Arguably the most interpretable metric is word accuracy, since it gives an idea about the number of lexical units that would have to be corrected, whereas MT metrics are less interpretable, given that they are designed to incorporate a certain degree of flexibility concerning word order, which is not relevant for the task of spelling normalisation. On the other hand, they have the advantage of penalising predictions that contain additional (hallucinated) tokens as well as correct tokens, a situation that is plausible given the use of sentence-level MT models.

We therefore choose to use a symmetrised version of word accuracy, which is the average between traditional word accuracy (aligning each gold token to predicted (sub)token(s)) and the reverse calculation (aligning each predicted token to gold (sub)token(s)).[8] More details on evaluation can be found in Appendix C. We also evaluate using MT metrics to test how they correlate with word accuracy.

5. Data

For training, development and test data, we present the FREEM corpus (short for *FREnch Early Modern*) called $\text{FREEM}_{\text{norm}}$.[9] The data covers a range of different genres of text throughout different decades of the 17th century, written in prose or verse, which have been semi-automatically normalised (Gabay et al., 2019) and manually corrected. Most of these texts belong to the *belles-lettres* (literature in its broadest sense), which is the type of source we want to normalise, but additional texts from different traditions (science, law, etc.) are present in the corpus. Some of the transcriptions have been produced specifically for this corpus and others have been borrowed from other projects: transcription rules are therefore not strictly equivalent from one text to another regarding, for instance old characters (e.g. *ſ*) or abbreviations (e.g. *õ*→*on*). "Normalisation" is understood here as a partial alignment with contemporary French: in some specific cases, specific spellings are maintained to keep the meter of the verse intact (e.g. the adverbial *-s*: *jusques*+vowel→ *jusques* and not *jusqu'* to maintain the three syllables).

The dataset has been split into train, dev and test sets, for which basic statistics can be found in Table 1. The split was done such that the test set contains a variety of different genres and periods (see Tables 7 and 5 in Appendix B), some of which are covered in the train and dev set and some of which are unseen.

In terms of the difficulty of the task, although many words remain unchanged between the original Modern French and their contemporary French normalisations (75.7% of all words in the training set), there are a non-negligible number of tricky cases. There are a large number of out-of-vocabulary (OOV) items in both the dev and test sets with respect to the training set, and approximately 0.3% of tokens are ambiguous (i.e. they correspond to several possible normalisations depending on the context). Aside minor differences such as punctuation (which is nevertheless not arbitrary, since it can be determined by context), capitals and accents, there are some interesting cases, such as ambiguity concerning verbal conjugations, which may require more contextual information (see Table 2 for two examples). For these cases, it is necessary to normalise words whilst taking into account their context (as in traditional MT). This justifies processing whole sentences rather than isolated words.

6. Experiments

6.1. Baselines

We compare the approaches described in Section 3 with two baseline approaches, the identity function and a basic rule-based approach.

Identity function This keeps the text unchanged.

Rule-based This is a stronger baseline comprising several dozen regular expressions, which were manually written based on simple corpus statistics from our training set. They range from purely typographic rules, which reflect the evolution of the writing system, to lexical rules, which reflect the evolution of the language. Here are a few examples, ordered from purely typographic to fully lexical:

- *ſ* → *s*, *õ* → *om* if followed by *m*, *b* or *p*, or *on* otherwise;
- *i* → *j* at the beginning of a word when followed by a vowel other than *i*;
- *estoit* → *était* and *estoient* → *étaient*.

In addition, we also assess the impact of the lexicon-based post-processing step on these baselines.

6.2. Experimental setup

All NMT models are trained using Fairseq (Ott et al., 2019), with default parameters unless otherwise specified. All models are trained until convergence; the best checkpoint is chosen based on symmetrised word accuracy on the dev set. Subword segmentation is applied using SentencePiece (Kudo and Richardson, 2018) and the BPE strategy (Sennrich et al., 2016).

We train SMT models using Moses (Koehn et al., 2007) and language models using KenLM (Heafield, 2011). We tune using `kbmira` to maximise BLEU.

[8] We first perform character-level alignment using Levenshtein and then realign on the token level with respect to the tokenisation of the gold and predicted sequences respectively.

[9] https://doi.org/10.5281/zenodo.5865428

Set	#sentences.	#tokens ModFr	#tokens Fr	#unique tokens ModFr	#unique tokens Fr	#OOV ModFr	#OOV Fr
Train	17,930	264,311	263,669	21,329	17,238	-	-
Dev	2,443	40435	40294	6736	5993	1,766	1,312
Test	5,706	86,432	86,211	10,457	8,915	3,596	2,530

Table 1: Statistics for the FREEM$_{norm}$ corpus for Modern French (ModFr) and contemporary French (Fr). Texts are tokenised using the Moses tokeniser (Koehn et al., 2007) to calculate statistics and #OOV corresponds to the number of unique out-of-vocabulary tokens.

	Normalisation example 1	Normalisation example 2
nostre 'our'	quel malheur est le **nôtre** 'what woe is **ours**'	Les larmes sont trop peu pour pleurer **notre** mal 'The tears are too few to cry (for) **our** pain'
appellez 'call'	N'**appelez** point des yeux le Galant à votre aide 'Do not **call** the Galant for help with your eyes'	...Royaumes, par nous vulgairement **appelés** Siam '...kingdoms, **known** popularly by us as Siam'

Table 2: Two examples of context-dependent ambiguity (Modern French words *nostre* and *appellez*) when normalising to contemporary French.

6.3. Best Model Search

6.3.1. Neural models

For LSTM and Transformer models, we performed hyper-parameter searches to maximise the symmetrised word accuracy on the development set. We explored (i) the network size (cf. Table 3 for LSTM models and Table 4 for Transformer models), (ii) the degree of subword segmentation via different BPE vocabulary sizes (500 1k, 2k, 4k, 8k, 16k, 24k), (iii) the learning rate (0.0005, 0.001, 0.001) and (iv) the batch size (1000, 2000, 3000, 4000 tokens). In order to avoid having to explore the combination of all parameters, we explored hyper-parameters in a step-wise fashion from (i) to (iv), keeping the best parameters from the previous step. We then explored variations on the network size parameters, varying attributes one below and one above the default values. Results were calculated as an average of three differently seeded runs for each combination. We began with default values for all hyper-parameters and varied only those mentioned.

Both models performed best with a BPE vocabulary of 1k, batch size of 3000 and learning rate of 0.001. The best network sizes were M for the LSTM, and a variant of the M model for the Transformer, with only 2 encoder layers rather than 4.

6.3.2. Statistical MT model

As for the neural models, we test several different granularities of segmentation: character-based, 500, 1k and 2k.[10] We use a 4-gram language model trained on the target side of either the parallel training data or the normalised texts of the FREEM$_{max}$ corpus (Gabay et al., 2022). The best subword segmentation is with vocabulary size 500 (interestingly not character-based as what has previously been used) and with the language model trained on the target side of the parallel training data.

[10]Larger vocabulary sizes result in worse scores and were also more difficult to train because of memory problems.

Size	#enc. layers	#dec. layers	embed. dim.
XS	1	1	128
S	2	2	256
M	3	3	384
L	4	4	512

Table 3: Network sizes explored for LSTM models.

Size	#attn. heads	#layers enc.	#layers dec.	Dim. embed.	Dim. ffwd.
S	2	2	2	128	512
M	4	4	4	256	1024
L	8	6	6	512	2048

Table 4: Network sizes explored for Transformer models. L corresponds to Transformer-base.

7. Results

We compare the approaches described in Section 3 according to the three evaluation metrics discussed in Section 4: symmetrised word accuracy (written as WordAcc), BLEU and CHRF.

Results are shown in Table 5. For MT approaches, we run each model three times with three random seeds and report the average score and standard deviation. Models (1)-(4) are baselines and already achieve relatively high scores. This is unsurprising, given the large number of words that do not need modifying: the identity function (copying the source text) gives 72.73% word accuracy. The rule-based approach is significantly better than the first baseline, and adding the post-processing step (+L*ffff*) considerably improves both results. The two statistical approaches, the hybrid ABA and SMT, both perform better than the baselines, with SMT actually performing the best out of all approaches. The NMT models perform slightly worse according to all metrics than SMT. Although the scores

	Model	WordAcc (%)	BLEU	ChrF	OOV WordAcc (%)
Baseline models					
(1)	Identity	72.73	40.25	73.77	43.00
(2)	Identify + Le*fff*	86.12	66.78	87.40	64.84
(3)	Rule-based	89.05	72.47	89.94	60.22
(4)	Rule-based + Le*fff*	90.85	76.90	91.70	66.51
Alignment-based approach					
(5)	ABA	95.14	87.70	95.84	69.50
MT approaches					
(6)	SMT	**97.10**±0.02	**92.59**±0.05	**97.71**±0.01	75.64±0.18
(7)	LSTM	96.14±0.08	91.77±0.21	96.85±0.08	**76.69**±0.70
(8)	TRANSFORMER	95.89±0.07	91.30±0.08	96.65±0.05	75.73±0.38
+ Lexicon-based post-processing					
(9)	ABA + Le*fff*	95.44	88.37	96.13	73.54
(10)	SMT + Le*fff*	**97.24**±0.02	**92.97**±0.05	**97.85**±0.01	**78.37**±0.20
(11)	LSTM + Le*fff*	96.25±0.10	92.07±0.25	96.95±0.10	**78.35**±0.79
(12)	TRANSFORMER + Le*fff*	96.01±0.09	91.62±0.14	96.76±0.08	77.51±1.00

Table 5: Results on the test set. "+ Le*fff*" indicates that the lexicon-based post-processing was applied.

Table 6: Word accuracy on OOV target tokens (%)

of LSTM and TRANSFORMER are very similar, LSTM scores are slightly higher. It is an interesting finding that the SMT outperforms NMT in our scenario, as this goes against recent findings for Modern French (Gabay and Barrault, 2020), despite us having more parallel data available. As for the baselines, adding the post-processing step improves both statistical and neural models, with the best result being SMT+Le*fff* with a symmetrised word accuracy of 97.24%.

As recommended by Robertson and Goldwater (2018), we also calculate word accuracy for OOV tokens (based on the gold tokens). Results (Table 6) show that the highest scoring model for OOV accuracy is LSTM, although if post-processing is applied, both SMT and LSTM show similar scores. Adding the post-processing step significantly helps the OOV accuracy of all methods, showing that it is an important complementary step.

The three evaluation metrics reveal the same pattern in results for these models, with BLEU varying more in absolute scores than the other metrics.

8. Comparative Analysis

8.1. How Similar are the Methods?

In Figure 2, we compare the predictions token by token and report the percentage of identical normalisations between methods.[11] Unsurprisingly, the neural methods (LSTM and TRANSFORMER) are most similar to each other. SMT is the most similar to TRANSFORMER and ABA is most similar to SMT.

8.2. Conservative or Zealous?

Depending on how the tool is to be applied, it can be better to have a more conservative or zealous model.

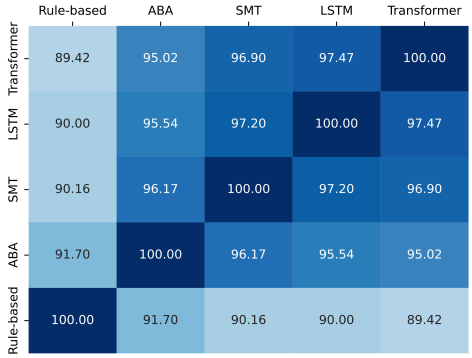

Figure 2: The percentage of identically normalised test set tokens between methods.

If automatic normalisation is to be used as a pre-annotation tool to help experts manually normalise texts, it is important for the automatic step not to introduce serious errors that could be more difficult to detect and time-consuming to correct. This is a concern notably for NMT-based models (Gabay and Barrault, 2020), which can be more creative in their transformation than either rule-based or SMT-based approaches. It may however be less of a problem if normalisation is to be used for certain downstream tasks using standard contemporary NLP tools (e.g. PoS-tagging or parsing). This is because a more zealous normalisation could provide better performance (by providing contemporary word forms), without the word forms themselves having to necessarily correspond to the correct ones.

To compare the methods for their conservativeness/zealousness, we align the output of each method with the source text and calculate (i) how often it changes a token that should have been kept as it is (Table 3), and (ii) how often it leaves a token untouched

[11] The analysis is computed against the first prediction for methods for which three random seeds were used.

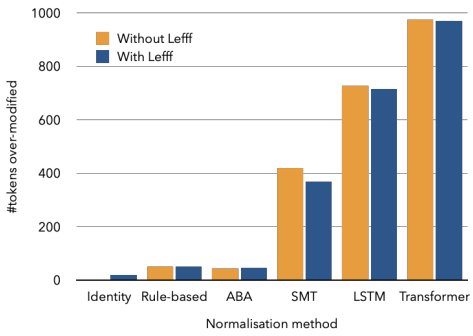

Figure 3: Comparison in the number of 'over-modified' test set tokens for each method.

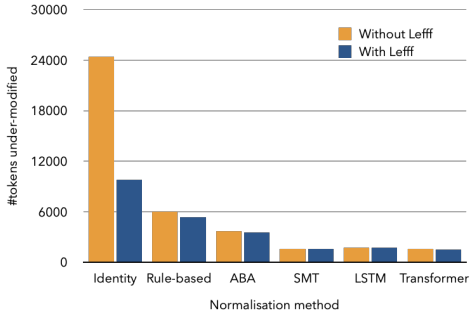

Figure 4: Comparison in the number of 'under-modified' test set tokens for each method.

that should be modified (Table 4). The identity function, rule-based system and ABA rarely over-modify, contrarily to SMT and NMT. Logically, the methods show the the inverse pattern for under-modification, with the identity and rule-based approaches being the most conservative and under-modifying the most. The SMT and NMT models under-modify at very similar rates, suggesting that performance differences could largely stem from over-modification rather than how much they under-modify. The best method, SMT, has the lowest rate of under-modification and a medium level of over-modification. ABA is interesting, because it under-modifies less than the baselines and yet does not over-modify as much as the MT approaches.

Adding the Le*fff*-based post-processing step has the effect of both correcting some over-modifications that were introduced and providing normalisations for previously unmodified tokens, thereby significantly improving the processing of OOV words.

8.3. Qualitative analysis of approaches

In this section, we compare the results of the best rule-based approach, ABA + Le*fff* and the best MT approach, SMT + Le*fff*, by using an alignment of the normalised versions of the dev data (available at `https://freem-corpora.github.io/models/norm_model/`).

Unsurprisingly, given that the substitution rules are not contextual, ABA + Le*fff* makes many errors in ambiguous cases, such as *A* instead of *À*, *prés* instead of *près*, *voila* instead of *voilà*, or *mes feux redoublez* instead of *mes feux redoublés*. Taking into account frequency scores either for the word replacement or for the character transformation rules in the training data may help avoid those mistakes. ABA + Le*fff* is also very sensitive to mistakes in the training corpus. For example, it succeeds in transforming *auoient* into *avaient* but not *avoient*, whereas SMT + Le*fff* succeeds. It also lacks some rules. For example it has no rule to normalise double consonants (for example *principalles* normalised into *principales*, *assouppit* into *assoupit*), whereas SMT + Le*fff* performs pretty well in this case. The SMT approach displays some more creative errors, but which appear easy spot if the normalised text is manually proof-read), e.g. *ma pẽſée* transformed into *ma pmentsée*. It is also prone to deleting certain words such as determiners, possibly because in some contexts they are less probable according to the language model. Finally, considering the fact that it is often the case that, when one of the two methods makes a mistake, the other one performs a correct normalisation, finding a relevant post-processing approach seems like a promising way to increase the quality of the results.

9. Conclusion

We have presented FreEM$_{\text{norm}}$, a new benchmark for the normalisation of Early Modern French, and compared a range of normalisation methods, including an alignment-based approach and various MT-based methods, with SMT outperforming all other approaches. Adding a post-processing with a contemporary French lexicon systematically helps, particularly for OOV tokens. We compare the strengths of the different methods, with rule- and alignment-based approaches being more conservative and MT approaches being less so. While MT approaches achieve the best accuracy, a model such as the alignment-based ABA is possibly more adapted to pre-annotation as it offers a good compromise between making good normalisation choices without overly normalising tokens that should not have been modified. We release all our data, models and scripts to encourage further research on this topic by the digital humanities community.

Acknowledgements

This work was performed using HPC resources from GENCI-IDRIS (Grant 20-AD011012254). It benefits from a State funding managed by the National Research Agency (ANR) under the Investments for the Future program (reference ANR-16-IDEX-0003, I-Site FUTURE, *Cité des dames, créatrices dans la cité*) in addition to the contributions of institutions and partners involved. It was also partly funded by the first and penultimate authors' chairs in the PRAIRIE institute funded by the French national agency ANR as part of the "Investissements d'avenir" programme under the reference ANR-19-P3IA-0001.

10. Bibliographical References

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*, San Diego, CA, USA.

Baron, A. and Rayson, P. (2009). Automatic standardisation of texts containing spelling variation: How much training data do you need? In *Proceedings of the Corpus Linguistics Conference: CL2009*, University of Liverpool, UK.

Bollmann, M. and Søgaard, A. (2016). Improving historical spelling normalization with bi-directional LSTMs and multi-task learning. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics*, pages 131–139, Osaka, Japan.

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.

Bollmann, M. (2012). (Semi-)automatic normalization of historical texts using distance measures and the Norma tool. In *Proceedings of the Second Workshop on Annotation of Corpora for Research in the Humanities (ACRH-2)*, pages 3–14, Lisbon, Portugal.

Bollmann, M. (2019). A large-scale comparison of historical text normalization systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3885–3898, Minneapolis, Minnesota.

Domingo, M. and Casacuberta, F. (2018a). A Machine Translation Approach for Modernizing Historical Documents Using Back Translation. In *Proceedings of the 15th International Workshop on Spoken Language Translation (IWSLT 2018)*, pages 38–47, Bruges, Belgium.

Domingo, M. and Casacuberta, F. (2018b). Spelling Normalization of Historical Documents by Using a Machine Translation Approach. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 129–137, Alicante, Spain.

Domingo, M. and Casacuberta, F. (2021). A comparison of character-based neural machine translations techniques applied to spelling normalization. In Alberto Del Bimbo, et al., editors, *Pattern Recognition. ICPR International Workshops and Challenges*, pages 326–338, Cham. Springer International Publishing.

Domingo, M., Chinea-Rios, M., and Casacuberta, F. (2017). Historical documents modernization. *The Prague Bulletin of Mathematical Linguistics*, 108:295–306.

Fix, H., (1980). *Automatische Normalisierung - Vorarbeit zur Lemmatisierung eines diplomatischen altisländischen Textes*, pages 92–100. Max Niemeyer Verlag.

Fourrier, C., Bawden, R., and Sagot, B. (2021). Can cognate prediction be modelled as a low-resource machine translation task? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 847–861, Online.

Gabay, S. and Barrault, L. (2020). Traduction automatique pour la normalisation du français du XVIIe siècle. In *Actes de la 6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition), Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 2 : Traitement Automatique des Langues Naturelles*, pages 213–222, Nancy, France.

Gabay, S., Riguet, M., and Barrault, L. (2019). A Workflow For On The Fly Normalisation Of 17th c. French. In *Proceedings of the 2019 Digital Humanities Conference*, Utrecht, Netherlands.

Gabay, S. (2021). Beyond Idiolectometry? On Racine's Stylometric Signature. In Maud Ehrmann, et al., editors, *Conference on Computational Humanities Research 2021*, pages 359–376, Amsterdam, Netherlands.

Hämäläinen, M., Säily, T., Rueter, J., Tiedemann, J., and Mäkelä, E. (2018). Normalizing early English letters to present-day English spelling. In *Proceedings of the Second Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 87–96, Santa Fe, New Mexico.

Hauser, A. W. and Schulz, K. U. (2007). Unsupervised Learning of Edit Distance Weights for Retrieving Historical Spelling Variations. In *Proceedings of the First Workshop on Finite-State Techniques and Approximate Search*, pages 1—6.

Heafield, K. (2011). KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Kogkitsidou, E. and Gambette, P. (2020). Normalisation of 16th and 17th century texts in French and geographical named entity recognition. In *Proceedings of the 4th ACM SIGSPATIAL Workshop on Geospatial Humanities*, pages 28–34, Seattle, Washington,

USA.

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

Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium.

Ljubesic, N., Zupan, K., Fiser, D., and Erjavec, T. (2016). Normalising Slovene data: historical texts vs. user-generated content. In Stefanie Dipper, et al., editors, *Proceedings of the 13th Conference on Natural Language Processing, KONVENS 2016*, volume 16 of *Bochumer Linguistische Arbeitsberichte*, pages 146–155, Bochum, Germany.

Maurel, D., Friburger, N., Antoine, J.-Y., Eshkol, I., and Nouvel, D. (2011). Cascades de transducteurs autour de la reconnaissance des entités nommées. *Traitement automatique des langues*, 52(1):69–96.

Mitankin, P., Gerdjikov, S., and Mihov, S. (2014). An approach to unsupervised historical text normalisation. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, DATeCH '14, pages 29–34, Madrid, Spain.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota, USA.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Pettersson, E., Megyesi, B., and Nivre, J. (2013a). Normalisation of historical text using context-sensitive weighted Levenshtein distance and compound splitting. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NoDaLiDa 2013)*, pages 163–179, Oslo, Norway.

Pettersson, E., Megyesi, B., and Tiedemann, J. (2013b). An SMT Approach to Automatic Annotation of Historical Text. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NoDaLiDa 2013)*, pages 54–69, Oslo, Norway.

Pettersson, E., Megyesi, B., and Nivre, J. (2014). A Multilingual Evaluation of Three Spelling Normalisation Methods for Historical Text. In *Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 32–41, Gothenburg, Sweden.

Piotrowski, M. (2012). *Natural language processing for historical texts*, volume 5(2) of *Synthesis lectures on human language technologies*. Morgan & Claypool Publishers.

Popović, M. (2015). chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal.

Porta, J., Sancho, J.-L., and Gomez, J. (2013). Edit Transducers for Spelling Variation in Old Spanish. In *Proceedings of the Workshop on Computational Historical Linguistics (NoDaLiDa 2013)*, pages 70–79, Oslo, Norway.

Reynaert, M., Hendrickx, I., and Marquilhas, R. (2012). Historical spelling normalization. A comparison of two statistical methods: TICCL and VARD2. In *Proceedings of the Second Workshop on Annotation of Corpora for Research in the Humanities (ACRH-2)*, Lisbon, Portugal.

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

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

Sennrich, R. and Zhang, B. (2019). Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Tang, G., Cap, F., Pettersson, E., and Nivre, J. (2018). An Evaluation of Neural Machine Translation Models on Historical Spelling Normalization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1320–1331, Santa Fe, New Mexico, USA.

Tjong Kim Sang, E., Bollmann, M., Boschker, R., Casacuberta, F., Dietz, F., Dipper, S., Domingo, M., van der Goot, R., van Koppen, M., Ljubešić, N., Östling, R., Petran, F., Pettersson, E., Scherrer, Y., Schraagen, M., Sevens, L., Tiedemann, J., Vanalle-

meersch, T., and Zervanou, K. (2017). The CLIN27 shared task: Translating historical text to contemporary language for improving automatic linguistic annotation. *Computational Linguistics in the Netherlands Journal*, 7:53–64.

Trieu, H. L., Tran, D.-V., and Le Nguyen, M. (2017). Investigating phrase-based and neural-based machine translation on low-resource settings. In *Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation*, pages 384–391, Cebu City, Philippines.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. U., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008.

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

11. Language Resource References

ATILF. (2019). Morphalou. https://hdl.handle.net/11403/morphalou/v3.1. ORTOLANG (Open Resources and TOols for LANGuage).

CasEN Team. (2019). Casen 1.4. https://tln.lifat.univ-tours.fr/medias/fichier/casen-fr-1-4_1596032302677-zip?ID_FICHE=332027&INLINE=FALSE.

Dipper, S. and Schultz-Balluff, S. (2013). The Anselm Corpus: Methods and perspectives of a parallel aligned corpus. In *Proceedings of the workshop on computational historical linguistics at NoDaLiDa 2013*, pages 27–42, Oslo, Norway.

Erjavec, T., Ringlstetter, C., Žorga, M., and Gotscharek, A. (2011). A lexicon for processing archaic language: the case of XIXth century Slovene. In *First International Workshop on Lexical Resources*.

Gabay, S., Bartz, A., Chagué, A., and Gambette, P. (2022). FreEM max. https://github.com/FreEM-corpora/FreEMmax_OA.

Ortiz Suarez, P., Gabay, S., Bartz, A., Bawden, R., Sagot, B., and Gambette, P. (2022). From FreEM to D'AlemBERT: a Large Corpus and a Language Model for Early Modern French. In *Proceedings of the 13th International Conference on Language Resources and Evaluation (LREC'22)*, Marseille, France.

Prolex Team. (2013). Prolex 1.2. https://tln.lifat.univ-tours.fr/medias/fichier/prolex-unitex-1-2_1562935068094-zip?ID_FICHE=321994&INLINE=FALSE.

Riguet, M. (2019). Normalisa, Script à base de règles pour normaliser les textes français du XVI^{e} au XIX^{e} siècle. https://github.com/mriguet/Normalisa/.

Sagot, B. (2010). The Le*fff*, a Freely Available and Large-coverage Morphological and Syntactic Lexicon for French. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta.

The French Wikisource Community. (2022). Wikisource:dictionnaire. https://fr.wikisource.org/wiki/Wikisource:Dictionnaire.

A. ABA Normalisation Rules

The character transformation rules used in the second step of ABA include ſ → s, ß → ss, & → et; the resolution of letters with a tilde used to abbreviate an n or an m; sç → s; final oing → oin; final y → i; sch → ch; aye → aie, oye → oie. The obtained word is considered as an initial candidate followed by the supplementary candidates obtained with the following rules: ct → t; vowel followed by dv → same vowel followed by v; final ans → ands, final ens → ends, final ans → ants, final ens → ents; final ois → ais (same with oit and oient); final ez → és, final és → ez; st → t, est → ét; as followed by m n q or t → â followed by the same letter (same with es, is, os and us); y → i; ü or eü → u. Finally, for all generated candidates, the following transformation rules are applied: is → î, ai → aî, u → v, v → u, non final e not followed by s → é.

B. Distribution of the Datasets by Decade and Genre

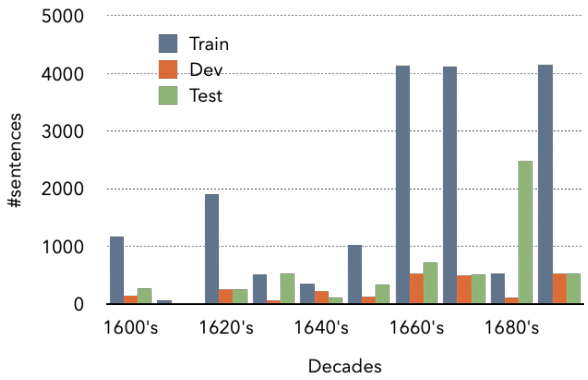

Figure 5: Distributions of data in terms of decades.

Genre	Train	Dev	Test
Caractères	190	25	25
Comédie	4870	619	623
Tale	120	15	15
Correspondence	1533	198	199
Law	61	0	0
Fables	899	112	114
Journalism	142	0	0
Medicine	0	59	114
Philosophy	455	57	200
Physics	0	0	182
Poetry	1777	224	226
Novel	1071	132	730
Memoir novel	213	27	27
Theology	560	70	72
Tragedy	5847	708	3155
Travel	192	24	24

Table 7: Number of sentences per genre.

C. Evaluation details

Word accuracy is calculated by aligning the set of sentences (each reference sentences and its normalised sentence) on the character level and then using the alignment matrix to produce a token-level alignment.

Initial Character-level Alignment Character-level alignment is performed using a modified (weighted) version of Levenshtein, whereby certain characters are considered equivalent (e.g. accented and non-accented versions of characters, long s (ſ) and s). The alignment is also designed to avoid tokenisation and punctuation mismatches unless they are really necessary for a successful alignment:

- by default, the cost of a substitution is 1, whereas the cost of an insertion or a deletion is 0.8;
- the cost of a substitution of a reference white-space character with a non-white-space is prohibitive (1,000,000);
- the cost of a substitution of a reference non-white-space character with a white-space is 30;
- the cost of a substitution involving a punctuation mark (within ,.;-!?') is 20;
- the cost of the deletion of a white-space character in the reference is prohibitive;
- the cost of the insertion of a white-space character in the reference is 2.

Token-level alignment The token-level alignment must necessarily be carried out with respect to the tokenisation of one of the sequences (there is not always a one-to-one mapping between reference and normalised tokens). We carry out tokenisation prior to character-level alignment using a very basic tokeniser lightly adapted to French (breaking on whitespace and around punctuation) and use then use whitespace tokens to delimit tokens when token-aligning the two sequences. We can either take the tokenisation of the reference sequence or of the normalised sequence as the basis for alignment. We preserve information about token boundaries such that different segmentations will be penalised even if the non-whitespace characters are identical.

(1) Ref: surtout j'ai choisi davantage ses écrits
MT: sur tout ji choisi d'avantage ses escrits,

(2) Align: surtout|||sur␣␣␣␣tout j'|||j
ai|||i choisi davantage|||d'␣␣avantage ses écrits|||escrits

For example, given a reference (Ref) and a predicted normalisation (MT) as shown in Example 1, the alignment in Example 2 is produced, where:

- ||| indicates that the reference and MT output do not match for that token;

- ▁ indicates that there is a token boundary introduced by the tokeniser in the aligned sequence of characters. Where there is also a space in the original sequence (before tokenisation), a double ▁▁ is indicated (case of over-merging);
- ▒ indicates that there is no token boundary to the right (case of over-splitting).

Symmetrised Accuracy Once aligned, the accuracy is the number of tokens for which the corresponding token is identical divided by the total number of tokens. We calculate a symmetrised accuracy, which is the average between the two accuracies: (i) the reference sentences are used as the basis for alignment and (ii) the normalised sentences are used as the basis for alignment. This is important because it helps to penalise very poor normalisations, such as those that can be produced by some MT-style models, where words can be hallucinated. If the accuracy is only computed according to the reference tokenisation, it is possible for all hallucinated words to be aligned to a single reference token and therefore penalised very little with respect to the amount of noise added.