Pattern Analysis and Applications
https://doi.org/10.1007/s10044-023-01164-w

ORIGINAL ARTICLE

Interactive machine translation for the language modernization and spelling normalization of historical documents

Miguel Domingo[1] · **Francisco Casacuberta**[1]

Received: 30 August 2022 / Accepted: 21 March 2023

Abstract
Historical documents are an important part of our cultural heritage. Among other task related to their processing, it is important to modernize their language in order to make them accessible to a broader audience and to achieve an orthography consistency to reduce the linguistic variation inherent in them. Language modernization and spelling normalization have those goals in mind. However, they still have a long way to go. Thus, in order to help scholars generate error-free modernizations/normalizations when the quality is essential, we propose an interactive framework based on interactive machine translation. In this work, we deployed two different interactive protocols into these tasks. We evaluated our proposal under simulated environments, observing significant reductions of the human effort.

Keywords Interactive machine translation · Language modernization · Spelling normalization · Historical documents

1 Introduction

Historical documents possess an outstanding cultural value. They are a unique public asset, forming the collective and evolving memory of our societies [44]. For this reason, with the aim of converting these documents to digital form, many tasks revolve around the processing of historical documents.

One of such task is language modernization. Due to the evolving nature of human language, historical documents are mostly limited to scholars. Thus, in order to make these documents available to a broader audience, language modernization aims to automatically generate a new version of a given document written in the modern version of its original language. However, while it succeeds in helping non-experts to understand the content of a historical document, language modernization is not error-free.

Similarly, another task related to the processing of historical documents is spelling normalization. Besides the evolving nature of human language, spelling conventions were not created until recently. Therefore, orthography changes depending on the author and time period, which could lead to an astonishing variety for writing a given word (e.g., Laing [28] pointed out more than 500 different forms recorded for writing the preposition *through*). These linguistic variations are present in historical documents and have always been a concern for scholars in humanities [6]. Spelling normalization tackles this problem by adapting a document's spelling to modern standards. However, it is still not able to produce error-free normalizations.

In both cases, scholars need to correct the system's outputs in those cases in which error-free modernized/normalized versions are needed. With the aim to help scholars to generate these error-free versions, we propose to deploy the interactive machine translation (IMT) collaborative framework into these tasks. In this methodology, a human and a translation system work together to produce the final translation.

This work builds upon Domingo and Casacuberta [14], which applied the IMT framework to language modernization. Our contributions are as follows:

- Further study the integration of prefix-based and segment-based IMT into language modernization.
- Integration of prefix-based and segment-based IMT into spelling normalization.

✉ Miguel Domingo
midobal@prhlt.upv.es

Francisco Casacuberta
fcn@prhlt.upv.es

[1] PRHLT Research Center, Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain

Published online: 24 April 2023

2 Related work

While it has been manually applied to the literature for centuries (e.g., *The Bible* has been adapted and translated for generations in order to preserve and transmit its contents [21]), automatic language modernization is a young research field. A shared task for translating historical text to contemporary language [54] was one of the first related works. However, although they approached language modernization using a set of rules, the task was focused on achieving an orthography consistency on the document's spelling. Domingo and Casacuberta [10] proposed a neural machine translation (NMT) approach. Sen et al. [47] augmented the training data by extracting pairs of phrases and adding them as new training sentences. Domingo and Casacuberta [13] proposed a method to profit from modern documents to enrich the neural models and conducted a user study. Lastly, Peng et al. [36] proposed a method for generating modernized summaries of historical documents.

Some approaches to spelling normalization include creating an interactive tool that includes spell checking techniques to assist the user in detecting spelling variations [3]. There is a combination of a weighted finite-state transducer, combined with a modern lexicon, a phonological transcriber and a set of rules [40]. There is a combination of a list of historical words, a list of modern words and character-based statistical machine translation (SMT) [46]. A multitask-learning approach using a deep bi-long short-term memory (LSTM) [23] is applied at a character level [7]. Ljubešic et al. applied a token/segment-level character-based SMT approach to normalize historical and user-created words [31]. Korgachina applied rule-based machine translation (RBMT), character-based machine translation (CBMT) and character-based neural machine translation (CBNMT) [27]. Domingo and Casacuberta [11] evaluated word-based and character-based MT approaches, finding character-based to be more suitable for this task and that SMT systems outperformed NMT systems. Tang et al. [53], however, compared many neural architectures and reported that the NMT models are much better than SMT models in terms of character error rate (CER). Finally, Hämäläinen et al. [22] evaluated SMT, NMT, an edit-distance approach, and a rule-based finite state transducer and advocated for a combination of these approaches to make use of their individual strengths.

The IMT framework was introduced during the *TransType* project [17] and was further developed during *TransType2* [4]. New contributions to this framework include developing new generations of the suffix [56] and profiting from the use of the mouse [45]. Marie et al. [32] introduced a touch-based interaction to iteratively improve translation quality. Lastly, Domingo et al. [15] introduced a segment-based protocol that broke the left-to-right limitation. With the rise of NMT, the interactive framework was deployed into the neural systems [25, 39], adding online learning techniques [38]; and reinforcement and imitation learning [29].

3 Approaches

In this section, we present and describe our different proposals to tackle language modernization and spelling normalization. All approaches rely on machine translation (MT), whose framework approximates a probability distribution using a mathematical model whose parameters are estimated from a collection of parallel data, in order to compute the translation probability (Pr) of the target sentence given a source sentence.

Thus, given a source sentence x_1^J, MT aims to find the most probable translation $\hat{y}_1^{\hat{I}}$ [8]:

$$\hat{y}_1^{\hat{I}} = \arg\max_{y_1^I} \Pr\left(y_1^I | x_1^J\right) \tag{1}$$

3.1 Language modernization

We confront language modernization from an MT perspective: The language of the original document would be the *source language*, and the modernized language would be the *target language*. With this in mind, we propose two different approaches based on SMT and NMT.

3.1.1 SMT approach

This approach is based on SMT, which uses models that rely on a log-linear combination of different models [34]. For years, this has been the prevailing approach to compute Eq. (1). Among others, it mainly combines phrase-based alignment models, reordering models and language models [58].

Given a parallel corpus in which for each original document its modernized version (parallel at a line level) is also available, this approach tackles language modernization as a conventional translation task: We train an SMT system using the original documents as the source part of the training data and their modernized versions as the target data.

3.1.2 NMT approaches

These approaches rely on NMT, which make use of neural networks to model Eq. (1). Its most frequent architecture is based on an encoder–decoder, featuring recurrent networks [2, 51], convolutional networks [19] or attention mechanisms [57]. At the encoding state, the source sentence is projected into a distributed representation. Then, at the decoding step, the decoder generates its most likely translation—word by word—using a beam search method [51]. The model parameters are typically estimated via stochastic gradient descent [43], jointly on large parallel corpora. Finally, the system obtains the most likely translation at decoding time by means of a beam search method.

Like the SMT approach, these approaches tackle language modernization as a conventional translation task but using NMT instead of SMT. Additionally, since the scarce availability of parallel training data is a frequent problem for historical data [7] and since NMT needs larger quantities of parallel training data than we have available (see Sect. 5.1), we followed Domingo and Casacuberta's [13] proposal for enriching the neural models with synthetic data: We apply feature decay algorithm (FDA) [5] to a monolingual corpus in order to filter it and obtain a more relevant subset. Then, we follow a back-translation approach [48] to train an inverse system—using the modernized version of the training dataset as source, and the original version as target. Following that, we translate the monolingual data with this system, obtaining a new version of the documents which, together with the original modern documents, conform the synthetic parallel data. After that, we train a NMT modernization system with the synthetic corpus. Finally, we fine-tune the system by training a few more steps using the original training data.

We made use of two different NMT modernization approaches, whose difference is the architecture of the neural systems:

- NMT_{LSTM}: This approach uses a recurrent neural network (RNN) [23] architecture with LSTM cells.
- $\text{NMT}_{\text{Transformer}}$: This approach uses a transformer [57] architecture.

3.2 Spelling normalization

We tackle spelling normalization similarly to language modernization (see Sect. 3.1). However, since in spelling normalization changes frequently occur at a character level, we followed a CBMT strategy. Due to spelling normalization being a much simpler problem than MT, we decided to use the simplest approach: splitting words into characters and considering each character as a token. Then, we consider the language of the original documents as the *source language* and its normalized version as the *target language*.

3.2.1 CBSMT approach

Like in language modernization's SMT approach (see Sect. 3.1.1), given a parallel dataset of historical documents and their normalized equivalents, this approach tackles spelling normalization as a conventional translation task—considering the document's language as the source language and its normalized version as the target language. In this case, however, we follow a character-based statistical machine translation (CBSMT) strategy: The document's words are split into characters and, then, conventional SMT is applied.

3.2.2 CBNMT approaches

These approaches are similar to the language modernization's NMT approaches, but using a CBNMT strategy to model Eq. (1). Additionally, since CBNMT also needs larger quantities of parallel training data than we have available (see Sect. 5.1), we followed Domingo and Casacuberta's [12] proposal for enriching the neural models with synthetic data: Given a collection of modern documents from the same language as the original document, we train a CBSMT system using the normalized version of the training dataset as source and the original version as target. We, then, use this system to translate the modern documents, obtaining a new version of the documents. This new version, together with the original modern document, conforms a synthetic parallel data which can be used as additional training data. After that, we combine the synthetic data with the training dataset, replicating several times the training dataset in order to match the size of the synthetic data and avoid over-fitting [9]. Finally, we use the resulting dataset to train the enriched CBNMT system.

Like in language modernization, we made use of two different CBNMT modernization approaches, whose difference is the architecture of the neural systems:

- $\text{CBNMT}_{\text{LSTM}}$: This approach uses a RNN architecture with LSTM cells.
- $\text{CBNMT}_{\text{Transformer}}$: This approach uses a transformer [57] architecture.

4 Interactive machine translation

In this work, we deploy the IMT framework into language modernization and spelling normalization. This framework proposes a collaborative process in which a human translator works together with an MT system to generate the final

translations. Thus, we can adapt it to language modernization and spelling normalization to create a collaborative framework between scholars and the modernization/normalization systems. In this section, we present and describe the two different IMT protocols we made use of: prefix-based and segment-based.

4.1 Prefix-based IMT

The prefix-based protocol proposes an iterative framework in which users correct the leftmost wrong word from a translation hypothesis, and the system generates a new hypothesis taking into account the user's feedback. Initially, the system proposes a translation hypothesis y_1^I of length I. The user, then, reviews this hypothesis and corrects the leftmost wrong word y_i. With this correction, they are inherently validating all the words that precede the corrected word, forming a validated prefix $\tilde{y}_1^i$, that includes the corrected word $\tilde{y}_i$. The system immediately reacts to this user feedback ($f = \tilde{y}_1^i$), generating a suffix $\hat{y}_{i+1}^I$ that completes $\tilde{y}_1^i$ to obtain a new translation of x_1^J : $\hat{y}_i^I = \tilde{y}_1^i\, \hat{y}_{i+1}^I$. This process is repeated until the user accepts the system's complete suggestion.

The suffix generation was formalized by Barrachina et al. [4] as follows:

$$\hat{y}_{i+1}^I = \arg\max_{I, y_{i+1}^I} \Pr\left(\tilde{y}_1^i y_{i+1}^I | x_1^J\right) \tag{2}$$

This equation is very similar to Eq. (1): At each iteration, the process consists in a regular search in the translations space but constrained by the prefix $\tilde{y}_1^i$.

Similarly, Peris et al. [39] formalized the neural equivalent as follows:

$$p\left(\hat{y}_{i'} \mid \hat{y}_1^{i'-1}, x_1^J, f = \tilde{y}_1^i; \Theta\right) = \begin{cases} \delta(\hat{y}_{i'}, \tilde{y}_{i'}), & \text{if } i' \leq i \\ \bar{\mathbf{y}}_{i'}^\top \mathbf{p}_{i'} & \text{otherwise} \end{cases} \tag{3}$$

where x_1^J is the source sentence; $\tilde{y}_1^i$ is the validated prefix together with the corrected word; Θ are the models parameters; $\bar{\mathbf{y}}_{i'}^\top$ is the one hot codification of the word i'; $\mathbf{p}_{i'}$ contains the probability distribution produced by the model at timestep i; and $\delta(\cdot, \cdot)$ is the Kronecker delta.

This is equivalent to a forced decoding strategy and can be seen as generating the most probable suffix given a validated prefix, which fits into the statistical framework deployed by Barrachina et al. [4].

4.2 Segment-based IMT

The segment-based protocol extends the human–computer collaboration defined in the previous protocol (see Sect. 4.1). Besides making a word correction, the user is now able to validate segments (sequences of words) and combine consecutive segments to create a larger one.

As in the prefix-based protocol, the process starts with the system suggesting an initial translation. The user, then, reviews it and validates those sequences of words which they consider to be correct. Then, they are able to delete words between validated segments to create a larger segment. After that, they make a word correction.

These three actions constitute the user feedback, which Domingo et al. [15] formalized as: $\tilde{\mathbf{f}}_1^N = \tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_N$; where $\tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_N$ is the sequence of N correct segments validated by the user in an interaction. Each segment is defined as a sequence of one or more target words. Therefore, each user action modifies the feedback differently:

1. Validating a new segment, inserting a new segment $\tilde{\mathbf{f}}_i$ in $\tilde{\mathbf{f}}_1^N$.
2. Merging two consecutive segments $\tilde{\mathbf{f}}_i$, $\tilde{\mathbf{f}}_{i+1}$ into a new one.
3. Introducing a word correction. This is introduced as a new one-word validated segment, $\tilde{\mathbf{f}}_i$, which is inserted in $\tilde{\mathbf{f}}_1^N$.

While the first two actions are optional—an iteration may not have new segments to validate—the last action is mandatory: It triggers the system to react to the user feedback, starting a new iteration of the process.

The system reacts to the user's feedback by generating a sequence of new translation segments $\hat{\mathbf{h}}_0^{N+1} = \hat{\mathbf{h}}_0, \ldots, \hat{\mathbf{h}}_{N+1}$. That means, an $\hat{\mathbf{h}}_i$ for each pair of validated segments $\tilde{\mathbf{f}}_i, \tilde{\mathbf{f}}_{i+1}$, being $1 \leq i \leq N$; plus one more at the beginning of the hypothesis, $\hat{\mathbf{h}}_0$; and another at the end of the hypothesis, $\hat{\mathbf{h}}_{N+1}$. The new translation of x_1^J is obtained by alternating validated and non-validated segments: $\hat{y}_1^I = \hat{\mathbf{h}}_0, \tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_N, \hat{\mathbf{h}}_{N+1}$. The goal is to obtain the best sequence of translation segments, given the user's feedback and the source sentence:

$$\hat{\mathbf{h}}_0^{N+1} = \arg\max_{h_0^{N+1}} \Pr\left(\mathbf{h}_0, \tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_N, \mathbf{h}_{N+1} \mid x_1^J\right) \tag{4}$$

This equation is very similar to Eq. (2). The difference is that, now, the search is performed in the space of possible substrings of the translations of x_1^J, constrained by the sequence of segments $\tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_N$, instead of being limited to the space of suffixes constrained by $\tilde{y}_1^i$.

Similarly, Peris et al. [39] formalized the neural equivalent of this protocol as follows:

$$p\left(y_{i_n+i'} \mid y_1^{i_n+i'-1}, x_1^J, f_1^N; \Theta\right) = \mathbf{y}_{i_n+i'}^\top \mathbf{p}_{i_n+i'} \tag{5}$$

where $f_1^N = f_1, \ldots, f_N$ is the feedback signal and $f_1, \ldots, f_N$ are a sequence of non-overlapping segments validated by the user; each alternative hypothesis y (partially) has the form $y = \ldots, f_n, h_n, f_{n+i}, \ldots; g_n$ is the non-validated segment;

$1 \leq i' \leq \hat{l}_n$; and l_n is the size of this non-validated segment and is computed as follows:

$$\hat{l}_n = \underset{0 \leq l_n \leq L}{\arg\max} \frac{1}{l_N + 1} \sum_{i'=i_n+1}^{i_n+l_n+1} \log p\left(y_{i'} \mid y_1^{i'-1}, x_1^J; \Theta\right) \tag{6}$$

5 Experimental framework

This section presents the details of our experimental session. We start by describing the corpora used for training our models. Then, we present the evaluation metrics used for assessing our proposal. After that, we detail the training procedure of our MT systems. Finally, we describe how we performed the user simulation.

5.1 Corpora

In our experimental session, we made use of the following corpora:

- **Language modernization**:
  - *Dutch Bible* [54]: A collection of different versions of the Dutch Bible. Among others, it contains a version from 1637—which we consider as the original version—and another from 1888—which we consider as the modern version (using nineteenth-century Dutch as if it were *modern Dutch*).
  - *El Quijote* [10]: the well-known seventeenth-century Spanish novel by Miguel de Cervantes, and its correspondent twenty-first-century version.
  - *OE-ME* [47]: contains the original eleventh-century English text *The Homilies of the Anglo-Saxon Church* and a nineteenth-century version—which we consider as *modern English*.
- **Spelling normalization**:
  - **Entremeses y Comedias**[1] [16]: A seventeenth-century Spanish collection of comedies by Miguel de Cervantes. It is composed of 16 plays, 8 of which have a very short length. Each line corresponds to the same line from its original manuscript.
  - **Quijote**[2] [16]: The seventeenth-century Spanish two-volumes novel by Miguel de Cervantes. Each line corresponds to the same line from its original manuscript.

Each corpus consists in a collection of historical documents and their correspondent versions in which either its language has been modernized or its spelling normalized. Therefore, each document contains two different versions whose content is parallel at a line level: the original document and its modernized/normalized counterpart.

Additionally, to enrich the neural models we made use of the following *modern documents*: the collection of Dutch books available at the *Digitale Bibliotheek voor de Nederlandse letteren*[3], for Dutch; and OpenSubtitles [30]—a collection of movie subtitles in different languages—for the rest of them. Table 1 contains the corpora statistics.

5.2 Evaluation metrics

We made use of the following well-known metrics in order to assess our proposal:

Word Stroke Ratio (WSR) [55] measures the number of words edited by the user, normalized by the number of words in the final translation.
Mouse Action Ratio (MAR) [4] measures the number of mouse actions made by the user, normalized by the number of characters in the final translation.

Additionally, to evaluate the initial quality of the modernization systems, we used the following well-known metrics:

BiLingual Evaluation Understudy (BLEU) [35] computes the geometric average of the modified n-gram precision, multiplied by a brevity factor that penalizes short sentences. In order to ensure consistent BLEU scores, we used *sacreBLEU* [41] for computing this metric.
Translation Error Rate (TER) [49]: computes the number of word edit operations (insertion, substitution, deletion and swapping), normalized by the number of words in the final translation. It can be seen as a simplification of the user effort of correcting a translation hypothesis on a classical post-editing scenario.

Finally, we applied approximate randomization testing (ART) [42]—with 10, 000 repetitions and using a p-value of 0.05—to determine whether two systems presented statistically significance.

5.3 MT systems

We trained SMT and CBSMT systems with *Moses* [26], following the standard procedure: We estimated a 5-gram language model—smoothed with the improved KneserNey

[1] https://users.pfw.edu/jehle/wcce.htm.
[2] https://users.pfw.edu/jehle/wcdq.htm.
[3] http://dbnl.nl/.

Table 1 Corpora statistics.

Language modernization							
		Dutch Bible		El Quijote		OE-ME	
		Original	Modernized	Original	Modernized	Original	Modernized
Train	S	35.2K		10K		2716	
	T	870.4K	862.4K	283.3K	283.2K	64.3K	69.6K
	V	53.8K	42.8K	31.7K	31.3K	13.3K	8.6K
Validation	S	2000		2000		500	
	T	56.4K	54.8K	53.2K	53.2K	12.2K	13.3K
	V	9.1K	7.8K	10.7K	10.6K	4.2K	3.2K
Test	S	5000		2000		500	
	T	145.8K	140.8K	41.8K	42.0K	11.9K	12.9K
	V	10.5K	9.0K	8.9K	9.0K	4.1K	3.2K
Modern documents	S	–	3.0M	–	2.0M	–	6.0M
	T	–	74.1M	–	22.2M	–	71.6M
	V	–	1.7M	–	211.7K	–	287.4K

Spelling normalization					
		Entremeses y Comedias		Quijote	
		Original	Normalized	Original	Normalized
Train	S	35.6K		48.0K	
	T	250.0	244.0K	436.0	428.0K
	V	19.0	18.0K	24.4	23.3K
Development	S	2.0K		2.0K	
	T	13.7	13.6K	19.0	18.0K
	V	3.0	3.0K	3.2	3.2K
Test	S	2.0K		2.0K	
	T	15.0	13.3K	18.0	18.0K
	V	2.7	2.6K	3.2	3.2K
Modern documents	S	–	500.0K	–	500.0K
	T	–	3.5M	–	3.5M
	V	–	67.3K	–	67.3K

S stands for number of sentences, T for number of tokens and V for size of the vocabulary. *Modern documents* refer to the monolingual data used to create the synthetic data. M denotes millions and K thousands

method—using *SRILM* [50], and optimized the weights of the log-linear model with MERT [33]. SMT systems were used both for the SMT modernization approach and for generating synthetic data to enrich the neural systems (see Sect. 3.1.2).

To built NMT and CBNMT systems, we used *NMT-Keras* [37]. We used long short-term memory units [20], with all model dimensions set to 512 for the RNN architecture. We trained the system using Adam [24] with a fixed learning rate of 0.0002 and a batch size of 60. We applied label smoothing of 0.1 [52]. At inference time, we used beam search with a beam size of 6. In order to reduce vocabulary, we applied joint byte pair encoding (BPE) [18] to all corpora, using 32, 000 merge operations.

For the transformer architecture [57], we used 6 layers; transformer, with all dimensions set to 512 except for the hidden Transformer feed-forward (which was set to 2048); 8 heads of transformer self-attention; 2 batches of words in a sequence to run the generator on in parallel; a dropout of 0.1; Adam [24], using an Adam beta2 of 0.998, a learning rate of 2 and Noam learning rate decay with 8000 warm up steps; label smoothing of 0.1 [52]; beam search with a beam size of 6; and joint BPE applied to all corpora, using 32, 000 merge operations.

5.4 User simulation

Due to the time and economic costs of conducting frequent human evaluations during the development stage, we conducted an evaluation with simulated users. These users had as goal to generate the modernizations/normalizations from the reference.

5.4.1 Prefix-based simulation

The simulation starts with the system offering an initial hypothesis. Then, the user compares it with the reference, looking for the leftmost wrong word. When they find it, they make a correction, validating a new prefix in the process. The cost associate to this correction is of one mouse action and one word stroke. After this, the system reacts to the user's feedback by generating a new suffix that completes the prefix to conform a new modernization/normalization hypothesis. This process is repeated until the hypothesis and the reference are the same.

To conduct this simulation, we used Domingo et al.'s [15] updated version of Barrachina et al.'s [4] software[4] for the SMT and CBSMT systems, and *NMT-Keras* [37]'s interactive branch for the NMT and CBNMT systems.

5.4.2 Segment-based simulation

For the sake of simplicity and without loss of generality, in this simulation we assumed that the user always corrects the leftmost wrong word and that validated word segments must be in the same order as in the reference. This assumption was also made by the original authors [15].

Like the previous simulation, the process starts with the system offering an initial hypothesis. Then, the user validates segments by computing the longest common subsequence [1] between this hypothesis and the reference. This has an associated cost of one action for each one-word segment and two actions for each multi-word segment. After this, the user checks if any pair of consecutive validated segments should be merged into a single larger segment (i.e., they appear consecutively in the reference but are separated by some words in the hypothesis). If there are, then they merge them, increasing mouse actions in one for each merge in which there was a single word between the segments or two otherwise. Finally, they correct the leftmost wrong word. Then, the system reacts to this feedback by generating a new hypothesis. This process is repeated until the hypothesis and the reference are the same.

To conduct this simulation, we made use of Domingo et al.'s [15] software[5] for the SMT and CBSMT systems, and *NMT-Keras*'s [37] interactive branch for the NMT and CBNMT systems.

6 Results

In this section, we present the results of the evaluation conducted for each task.

6.1 Language modernization

Table 2 presents the results of deploying the IMT framework into language modernization. It showcases the initial quality of each modernization system and compares their performance using the prefix-based or the segment-based framework.

The SMT approach obtained the best results by a large margin. The prefix-based protocol yields a reduction of the human effort of creating error-free modernizations. Additionally, the segment-based protocol obtains even larger reduction of the typing effort, at the expenses of a small increase in the use of the mouse—which is believed to have a smaller impact in the human effort [15].

Regarding the NMT approaches, despite that all of them yield a successful reduction of the human effort, these

[4] https://github.com/midobal/pb-imt.

[5] https://github.com/midobal/sb-imt.

Table 2 Experimental results of our language modernization IMT approaches.

Corpus	Approach	Modernization quality		Prefix-based		Segment-based	
		TER	BLEU	WSR	MAR	WSR	MAR
		[↓]	[↑]	[↓]	[↓]	[↓]	[↓]
Dutch bible	SMT	11.5	77.5	14.3	4.4	**9.0**	**10.8**
	NMT_{LSTM}	50.7†	43.4	42.6‡	9.2	42.6‡	50.9
	$NMT_{Transformer}$	50.3†	35.8	49.2‡	10.4	49.2‡	48.3
El Quijote	SMT	30.7	58.3	38.8	10.9	**22.0**	**19.7**
	NMT_{LSTM}	42.9	50.4	68.9‡	11.8	68.9‡	47.8
	$NMT_{Transformer}$	47.3	46.1	73.2‡	13.4	73.2‡	50.5
OE-ME	SMT	39.6	39.6	58.2	15.5	**28.2**	**26.1**
	NMT_{LSTM}	56.4	30.3	72.1‡	12.8†	72.1‡	59.5
	$NMT_{Transformer}$	58.9	28.2	73.5‡	13.3†	73.5‡	49.5

The initial modernization quality is meant to be a starting point comparison of each system. All results are significantly different between all approaches except those denoted with †. Given the same approach, all results are significantly different between the different IMT protocols except those denoted with ‡. [↓] indicates that the lowest the value the highest the quality. [↑] indicates that the highest the value the highest the quality. Best results are denoted in **bold**

Table 3 Experimental results of our spelling normalization IMT approaches.

Corpus	Approach	Normalization quality			Prefix-based		Segment-based	
		CER	TER	BLEU	KSR	MAR	KSR	MAR
		[↓]	[↓]	[↑]	[↓]	[↓]	[↓]	[↓]
Entremeses	CBSMT	1.3†	4.4	91.7	**0.9**‡	**4.1**	0.7‡	6.7
y	$CBNMT_{LSTM}$	3.5	9.4	84.9	1.9‡	2.1†	1.9‡	3.3
Comedias	$CBNMT_{Transformer}$	1.5†	6.5	87.2	1.4‡	2.1†	1.4‡	3.4
Quijote	CBSMT	2.5†	3.0†	94.4†	1.4†‡	3.7	1.1†‡	5.3
	$CBNMT_{LSTM}$	2.6†	4.3	93.9†	**1.4**†	**1.4**†‡	1.4†‡	2.1
	$CBNMT_{Transformer}$	2.2†	3.7†	94.4†	**1.5**†‡	**1.4**†	1.5†‡	2.1

The initial modernization quality is meant to be a starting point comparison of each system. All results are significantly different between all approaches except those denoted with †. Given the same approach, all results are significantly different between the different IMT protocols except those denoted with ‡. [↓] indicates that the lowest the value the highest the quality. [↑] indicates that the highest the value the highest the quality. Best results are denoted in **bold**

diminish significantly smaller than the ones obtained by the SMT approach. Furthermore, the segment-based protocol does not offer any benefit with respect to the prefix-based—both protocols have the same typing effort—while it has a significant increase in the mouse usage. Most likely, this is related to the system's modernization quality being smaller than the SMT system.

Finally, as already mentioned, it is worth noting the quality gap between the SMT and the NMT approaches—specially for the *Dutch Bible* dataset. While we created synthetic data to enrich the neural models (see Sect. 3.1.2), the scarce availability of historical training data is a known problem [7] that seems to have a bigger impact on the neural models, which have a tendency to need larger quantities of parallel training data. On the other hand, the SMT models need fewer resources and are capable of better exploiting the available data (specially given the particularities of this task).

6.2 Spelling normalization

Table 3 presents the results of deploying the IMT framework into spelling normalization. It presents the initial quality of each normalization system and compares their performance using the prefix-based or the segment-based protocol.

In the case of *Entremeses y Comedias*, the CBSMT approach yielded the best results for both protocols. For *Quijote*, all approaches had a similar behavior. When comparing protocols we observe that, while in all cases the IMT framework successfully reduced the human effort needed to generate error-free normalization, both protocols presented a similar typing effort. Most likely, this is due to the

Prefix-based

source (x): Ealle ðing he foresceawað and wát, and ealra ðeoda gereord he cann.

target translation ($\hat{y}$): All things he foresees and knows, and he understands the tongues of all nations.

IT-0	**MT**	All things he foresceawað and knows, and of all nations language he understands.
IT-1	**User**	All things he **foresees** and knows, and of all nations language he understands.
	MT	All things he foresees and knows, and of all nations language he understands.
IT-2	**User**	All things he foresees and knows, and **he** all nations language he understands.
	MT	All things he foresees and knows, and he understands of all nations language.
IT-3	**User**	All things he foresees and knows, and he understands **the** all nations language.
	MT	All things he foresees and knows, and he understands the beginning of all nations language.
IT-4	**User**	All things he foresees and knows, and he understands the **tongues** of all nations language.
	MT	All things he foresees and knows, and he understands the tongues all.
IT-5	**User**	All things he foresees and knows, and he understands the tongues **of**
	MT	All things he foresees and knows, and he understands the tongues of all.
IT-6	**User**	All things he foresees and knows, and he understands the tongues of all **nations.**
	MT	All things he foresees and knows, and he understands the tongues of all nations .
END	**User**	All things he foresees and knows, and he understands the tongues of all nations.

(a) The session starts with the system proposing an initial modernization. The user, then, looks for the leftmost wrong word and corrects it (*foresees* instead of *foresceawað*). Inherently, they are validating the prefix *All thing he*. Immediately, the system reacts to this feedback by suggesting a new hypothesis. The process is repeated until the user finds the system's hypothesis satisfactory.

Segment-based

source (x): Ealle ðing he foresceawað and wát, and ealra ðeoda gereord he cann.

target translation ($\hat{y}$): All things he foresees and knows, and he understands the tongues of all nations.

IT-0	**MT**	All things he foresceawað and knows, and of all nations language he understands.
IT-1	**User**	All things he **foresees** and knows, and of all nations language he understands.
	MT	All things he foresceawað foresees and knows, and language he understand of all nations .
IT-2	**User**	All things he foresees and knows, and he understand **the** of all nations .
	MT	All things he foresees and knows, and he understand the language of all nations .
IT-3	**User**	All things he foresees and knows, and he understand the **tongues** of all nations .
	MT	All things he foresees and knows, and he understand the tongues of all nations .
END	**User**	All things he foresees and knows, and he understands the tongues of all nations.

(b) The session starts with the system proposing an initial modernization. The user, then, reviews the hypothesis and selects all the word segments that considers to be correct (All things he , and knows , and and of all nations). Then they make a word correction (*foresees* instead of *foresceawað*). Immediately, the system reacts to this feedback by suggesting a new hypothesis. The process is repeated until the user finds the system's hypothesis satisfactory.

Fig. 1 Example in which both protocols successfully reduced the effort of generating error-free modernizations

highest initial quality of the systems: Since there are fewer errors to correct, using one methodology over the other one is not so relevant as when there are more errors. However, the segment-based protocol comes with a small increase in the mouse effort, since it has a more complex user protocol.

7 Qualitative analysis

In this section, we present a more in-depth study of the system's behaviors in the different tasks.

7.1 Language modernization

Figure 1 showcases an example in which the IMT framework significantly reduces the human effort of generating an error-free modernization of an old English document. While modernizing the sentence from scratch has a cost of 14 word strokes and one mouse action, and correcting the automatic modernization costs 7 word strokes and 7 mouse actions, the cost is reduced to 6 word strokes and 6 mouse actions using the prefix-based protocol, and 3 word strokes and 15 mouse actions—which have a smaller impact in the human effort—with the segment-based protocol.

Table 4 Statistics of the effort needed to generate the error-free modernizations.

Corpus	From scratch		Prefix-based		Segment-based	
	Word strokes	Mouse actions	Word strokes	Mouse actions	Word strokes	Mouse actions
Dutch Bible	140818	5000	20129	20129	13264	65690
El Quijote	48582	1650	18837	18837	11355	41237
OE-ME	15176	500	8835	8835	4641	18153

For the IMT protocols, only the SMT approach has been considered since it yielded the best results

Prefix-based

source (x): Pero no puedo pensar qué es lo que vio esta donzella en vuessa merced que assi la rindiesse y auassallasse.

target translation ($\hat{y}$): Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y avasallarla de ese modo.

IT-0	**MT**	Pero no puedo pensar qué es lo que vio esta doncella en vuestra merced que la rindiese y auassallasse.
IT-1	**User**	Pero no puedo **comprender** qué es lo que vio esta doncella en vuestra merced que la rindiese y auassallasse.
	MT	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced que la rindiese y auassallasse.
IT-2	**User**	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced **para** que la rindiese y auassallasse.
	MT	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para que la rindiese y auassallasse.
IT-3	**User**	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para **rendirla** la rindiese y auassallasse.
	MT	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y auassallasse que la.
IT-4	**User**	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y **avasallarla** que la.
	MT	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y avasallarla que.
IT-5	**User**	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y avasallarla **de**
	MT	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y avasallarla de .
IT-6	**User**	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y avasallarla de **ese**
	MT	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y avasallarla de ese modo.
END	**User**	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y avasallarla de ese modo.

(a) The session starts with the system proposing an initial modernization. The user, then, looks for the leftmost wrong word and corrects it (*comprender* instead of *pensar*). Inherently, they are validating the prefix *Pero no puedo*. Immediately, the system reacts to this feedback by suggesting a new hypothesis. The process is repeated until the user finds the system's hypothesis satisfactory.

Segment-based

source (x): Pero no puedo pensar qué es lo que vio esta donzella en vuessa merced que assi la rindiesse y auassallasse.

target translation ($\hat{y}$): Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y avasallarla de ese modo.

IT-0	**MT**	Pero no puedo pensar qué es lo que vio esta doncella en vuestra merced que la rindiese y auassallasse.
IT-1	**User**	Pero no puedo **comprender** qué es lo que vio esta doncella en vuestra merced que la rindiese y auassallasse.
	MT	Pero no puedo pensar comprender qué es lo que vio esta doncella en vuestra merced auassallasse y
IT-2	**User**	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced **para** y
	MT	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para la rindiese y auassallasse.
IT-3	**User**	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para **rendirla** rindiese y auassallasse.
	MT	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para la rindiese rendirla auassallasse y
IT-4	**User**	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y **avasallarla**
	MT	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y la rindiese auassallasse avasallarla
IT-5	**User**	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y avasallarla **de**
	MT	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y avasallarla de rindiese auassallasse
IT-6	**User**	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y avasallarla de **ese** auassallasse
	MT	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y avasallarla de rindiese auassallasse ese
IT-7	**User**	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y avasallarla de ese **modo.**
	MT	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y avasallarla de ese rindiese auassallasse modo.
IT-8	**User**	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y avasallarla de ese modo.
	MT	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y avasallarla de ese modo.
END	**User**	Pero no puedo comprender qué es lo que vio esta doncella en vuestra merced para rendirla y avasallarla de ese modo.

(b) The session starts with the system proposing an initial modernization. The user, then, reviews the hypothesis and selects all the word segments that considers to be correct (Pero no puedo , qué es lo que vio esta doncella en vuestra merced and y). Then they make a word correction (*comprender* instead of *pensar*). Immediately, the system reacts to this feedback by suggesting a new hypothesis. The process is repeated until the user finds the system's hypothesis satisfactory.

Fig. 2 Example of a case in which only the prefix-based protocol is able to reduce the human effort of generating error-free modernizations

Fig. 3 Example of normalizing the spelling of a sentence from *Entremeses y Comedias*

Prefix-based

source (x): ¿Será haçertado seguille?

target translation ($\hat{y}$): ¿Será acertado seguirle?

IT-0	**MT**	¿Será hacertado seguille?
IT-1	**User**	¿Será **a**acertado seguille?
	MT	¿Será a certado seguille?
IT-2	**User**	¿Será acertado segui **r**le?
	MT	¿Será acertado seguir le?
END	**User**	¿Será acertado seguirle?

(a) The session starts with the system proposing an initial hypothesis. The user, then, looks for the leftmost wrong character and corrects it (*a* instead of *h*). Inherently, they are validating the prefix *¿Será*. Immediately, the system reacts to this feedback by suggesting a new hypothesis. The process is repeated until the user finds the system's hypothesis satisfactory.

Segment-based

source (x): ¿Será haçertado seguille?

target translation ($\hat{y}$): ¿Será acertado seguirle?

IT-0	**MT**	¿Será hacertado seguille?
IT-1	**User**	¿Será h acertado segui **r** le?
	MT	¿Será h acertado seguir le?
END	**User**	¿Será acertado seguirle?

(b) The session starts with the system proposing an initial hypothesis. The user, then, reviews the hypothesis and selects all the character segments that considers to be correct (¿Será , acertado segui and le?). Then they make a character correction (˚ instead of *l*). Immediately, the system reacts to this feedback by suggesting a new hypothesis. The process is repeated until the user finds the system's hypothesis satisfactory.

Prefix-based

source (x): Mancha, y hechole confessar que es mas hermosa.

target translation ($\hat{y}$): Mancha, y héchole confesar que es más hermosa.

IT-0	**MT**	Mancha, y hechole confesar que es más hermosa.
IT-1	**User**	Mancha, y h **é**chole confesar que es más hermosa.
	MT	Mancha, y hé chóle confesar que es más hermosa.
IT-2	**User**	Mancha, y héch **o**le confesar que es más hermosa.
	MT	Mancha, y hécho le confesar que es más hermosa.
END	**User**	Mancha, y héchole confesar que es más hermosa.

(a) The session starts with the system proposing an initial hypothesis. The user, then, looks for the leftmost wrong character and corrects it (*é* instead of *e*). Inherently, they are validating the prefix *Mancha, y h*. Immediately, the system reacts to this feedback by suggesting a new hypothesis. The process is repeated until the user finds the system's hypothesis satisfactory.

Segment-based

source (x): Mancha, y hechole confessar que es mas hermosa.

target translation ($\hat{y}$): Mancha, y héchole confesar que es más hermosa.

IT-0	**MT**	Mancha, y hechole confesar que es más hermosa.
IT-1	**User**	Mancha, y h **é** chole confesar que es más hermosa.
	MT	Mancha, y hé chole confesar que es más hermosa.
END	**User**	Mancha, y héchole confesar que es más hermosa.

(b) The session starts with the system proposing an initial hypothesis. The user, then, reviews the hypothesis and selects all the character segments that considers to be correct (Mancha, y h and chole confesar que es más hermosa.). Then they make a character correction (*é* instead of *e*). Immediately, the system reacts to this feedback by suggesting a new hypothesis. The process is repeated until the user finds the system's hypothesis satisfactory.

Fig. 4 Example of normalizing the spelling of a sentence from *Quijote*

Figure 2 showcases an example in which only the prefix-based approach is able to reduce the human effort. Modernizing this old Spanish sentence from scratch has an associated cost of 21 word strokes and 1 mouse action, while post-editing the automatic modernization would cost 7 word strokes and 7 mouse actions. The prefix-based protocol is able to reduce the effort by 1 word stroke and 1 mouse action. However, the segment-based protocol maintains the typing effort while increasing the mouse effort to 28 mouse actions. This is due to a known weakness in this protocol, in which the system may fail to properly handle the user correction if they consist in out-of-vocabulary words.

Finally, Table 4 reflects the human effort needed to generate error-free modernizations. In all cases, the IMT framework significantly reduces the typing effort than generating the modernizations from scratch, at the cost of increasing the mouse effort[6] However, it is believed that the mouse has a smaller impact in the human effort [15].

Regarding the different IMT protocols, we can observe how the total mouse effort gets reduced by half with the segment-based protocol, while increasing the mouse effort by two times

[6] For the sake of simplicity, we have considered that only one mouse action per sentence is needed when generating the language modernizations from scratch.

Table 5 Statistics of the effort needed to generate the error-free normalizations.

Corpus	From scratch		Prefix-based		Segment-based	
	Key Strokes	Mouse Actions	Key Strokes	Mouse actions	Key strokes	Mouse actions
Entremeses y Comedias	59444	2000	546	546	437	3983
Quijote	83577	2000	1161	1161	892	4431

For the IMT protocols, only the SMT approach has been considered since no statistical difference has been observed between approaches

in the cases of *El Quijote* and *OE-ME*, and three times in the case of *Dutch Bible*. While these results have been obtained under a simulated environment, we believe that the effort reductions obtained by the segment-based protocol are significant enough to consider this protocol the most suitable for this task. Nonetheless, we would like to deepen in this study in a future work conducting a human evaluation, which would allow us to take into consideration other factors such as the time taken by each approach.

7.2 Spelling normalization

Figures 3 and 4 showcase some examples of generating error-free spelling normalizations using the interactive framework. As reflected in Table 3, all approaches and protocols yielded similar results. Since the systems have a high normalization quality, the orthography inconsistencies that need to be normalized typically consist in a few characters per sentence—with most sentences already yielding an error-free normalization.

Finally, Table 5 reflects the human effort needed to generate error-free modernizations. Like in the language modernization task, the IMT framework always succeeds in reducing the typing effort. Moreover, in this case the prefix-based protocol is also able to reduce the mouse effort, while the segment-based approach doubles the total number of mouse actions.

Overall, both IMT protocols yielded similar results. While the number of mouse actions in the segment-based protocol is considerably larger than in the prefix-based one, this difference is not statistically significant when normalizing by the number of characters (as reflected by the MAR metric at Table 3). Thus, while the prefix-based protocol seems to perform better on this task, a human evaluation—which would allow us to measure additional factors such as the time taken—needs to be conducted prior to arriving to a categorical conclusion.

8 Conclusions and future work

With the aim of helping scholars to generate error-free modernizations/normalizations, in this work we have deployed the interactive framework into two tasks related to the processing of historical documents: language modernization and spelling normalization. We deployed two different protocols to several MT modernization and normalization approaches.

Results show that the IMT framework always succeeded in reducing the human effort. For language modernization, the SMT approach yielded the best results under the segment-based protocol, reducing the typing effort in around two to ten points. In the case of spelling normalization, due to the high quality of the systems, all approaches and protocols behave similarly.

Finally, in a future work we would like to conduct a human evaluation with the help of scholars to better assess the benefits of applying the interactive framework to language modernization and spelling normalization.

Acknowledgements The research leading to these results has received funding from *Generalitat Valenciana* under project *PROMETEO/2019/121* and from *ValgrAI* (*Valencian Graduate School and Research Network for Artificial Intelligence*). We gratefully acknowledge *Andrés Trapiello* and *Ediciones Destino* for granting us permission to use their book in our research.

Data availability From the datasets used for the evaluation of this work, *Dutch Bible* and *OE-ME* are available from their original authors. *Entremeses y Comedias* and *Quijote* are not publicly available due to licensing restrictions but are available from the corresponding author on reasonable request. Finally, *El Quijote* is not publicly available due to licensing restrictions, and it is only available from the corresponding author on reasonable request and with the permission of *Andrés Trapiello* and *Ediciones Destino*.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

1. Apostolico A, Guerra C (1987) The longest common subsequence problem revisited. Algorithmica 2:315–336
2. Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate, arXiv:1409.0473
3. Baron A, Rayson P (2008) VARD2: a tool for dealing with spelling variation in historical corpora. In: Postgraduate conference in corpus linguistics
4. Barrachina S, Bender O, Casacuberta F et al (2009) Statistical approaches to computer-assisted translation. Comput Linguist 35:3–28
5. Biçici E, Yuret D (2015) Optimizing instance selection for statistical machine translation with feature decay algorithms. IEEE/ACM Trans Audio Speech Lang Process 23(2):339–350
6. Bollmann M (2018) Normalization of historical texts with neural network models. PhD thesis, Sprachwissenschaftliches Institut, Ruhr-Universität
7. Bollmann M, Søgaard A (2016) Improving historical spelling normalization with bi-directional lstms and multi-task learning. In: Proceedings of the international conference on the computational linguistics, pp 131–139
8. Brown PF, Pietra VJD, Pietra SAD et al (1993) The mathematics of statistical machine translation: parameter estimation. Comput Linguist 19(2):263–311
9. Chatterjee R, Farajian MA, Negri M, et al (2017) Multi-source neural automatic post-editing: Fbk's participation in the wmt 2017 ape shared task. In: Proceedings of the second conference on machine translation, pp 630–638
10. Domingo M, Casacuberta F (2018) A machine translation approach for modernizing historical documents using back translation. In: Proceedings of the international workshop on spoken language translation, pp 39–47
11. Domingo M, Casacuberta F (2018) Spelling normalization of historical documents by using a machine translation approach. In: Proceedings of the annual conference of the European association for machine translation, pp 129–137
12. Domingo M, Casacuberta F (2019) Enriching character-based neural machine translation with modern documents for achieving an orthography consistency in historical documents. In: Proceedings of the international workshop on pattern recognition for cultural heritage, pp 59–69
13. Domingo M, Casacuberta F (2020) Modernizing historical documents: a user study. Pattern Recognit Lett 133:151–157
14. Domingo M, Casacuberta F (2022) An interactive machine translation framework for modernizing the language of historical documents. In: Proceedings of the Iberian conference on pattern recognition and image analysis, pp 41–53
15. Domingo M, Peris Á, Casacuberta F (2017) Segment-based interactive-predictive machine translation. Mach Transl 31:1–23
16. F. Jehle F (2001) Works of Miguel de Cervantes in old- and modern-spelling. Indiana University Purdue University Fort Wayne
17. Foster G, Isabelle P, Plamondon P (1997) Target-text mediated interactive machine translation. Mach Transl 12:175–194
18. Gage P (1994) A new algorithm for data compression. C Users J 12(2):23–38
19. Gehring J, Auli M, Grangier D, et al (2017) Convolutional sequence to sequence learning. In: Proceedings of the international conference on machine learning, pp 1243–1252
20. Gers FA, Schmidhuber J, Cummins F (2000) Learning to forget: continual prediction with LSTM. Neural Comput 12(10):2451–2471
21. Given MD (2015) A discussion of bible translations and biblical scholarship. http://courses.missouristate.edu/markgiven/rel102/bt.htm
22. Hämäläinen M, Säily T, Rueter J, et al (2018) Normalizing early english letters to present-day english spelling. In: Proceedings of the workshop on computational linguistics for cultural heritage, social sciences, humanities and literature, pp 87–96
23. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
24. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980
25. Knowles R, Koehn P (2016) Neural interactive translation prediction. In: Proceedings of the association for machine translation in the Americas, pp 107–120
26. Koehn P, Hoang H, Birch A, et al (2007) Moses: open source toolkit for statistical machine translation. In: Proceedings of the annual meeting of the association for computational linguistics, pp 177–180
27. Korchagina N (2017) Normalizing medieval german texts: from rules to deep learning. In: Proceedings of the nordic conference on computational linguistics workshop on processing historical language, pp 12–17
28. Laing M (1993) The linguistic analysis of medieval vernacular texts: two projects at edinburgh'. In: Rissanen M, Kytd M, Wright S (eds) Corpora across the centuries: proceedings of the first international colloquium on english diachronic corpora, St Catharine's College Cambridge, pp 121–141
29. Lam TK, Schamoni S, Riezler S (2019) Interactive-predictive neural machine translation through reinforcement and imitation. In: Proceedings of machine translation summit, pp 96–106
30. Lison P, Tiedemann J (2016) Opensubtitles2016: extracting large parallel corpora from movie and tv subtitles. In: Proceedings of the international conference on language resources association, pp 923–929
31. Ljubešic N, Zupan K, Fišer D, et al (2016) Normalising slovene data: historical texts vs. user-generated content. In: Proceedings of the conference on natural language processing, pp 146–155
32. Marie B, Max A (2015) Touch-based pre-post-editing of machine translation output. In: Proceedings of the conference on empirical methods in natural language processing, pp 1040–1045
33. Och FJ (2003) Minimum error rate training in statistical machine translation. In: Proceedings of the annual meeting of the association for computational linguistics, pp 160–167
34. Och FJ, Ney H (2002) Discriminative training and maximum entropy models for statistical machine translation. In: Proceedings of the annual meeting of the association for computational linguistics, pp 295–302
35. Papineni K, Roukos S, Ward T, et al (2002) BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the annual meeting of the association for computational linguistics, pp 311–318
36. Peng X, Zheng Y, Lin C, et al (2021) Summarising historical text in modern languages. In: Proceedings of the conference of the european chapter of the association for computational linguistics, pp 3123–3142
37. Peris A, Casacuberta F (2018) NMT-Keras: a very flexible toolkit with a focus on interactive NMT and online learning. Prague Bull Math Linguist 111:113–124
38. Peris Á, Casacuberta F (2019) Online learning for effort reduction in interactive neural machine translation. Comput Speech Lang 58:98–126
39. Peris Á, Domingo M, Casacuberta F (2017) Interactive neural machine translation. Comput Speech Lang 45:201–220

40. Porta J, Sancho JL, Gómez J (2013) Edit transducers for spelling variation in old spanish. In: Proceedings of the workshop on computational historical linguistics, pp 70–79
41. Post M (2018) A call for clarity in reporting bleu scores. In: Proceedings of the third conference on machine translation, pp 186–191
42. Riezler S, Maxwell JT (2005) On some pitfalls in automatic evaluation and significance testing for mt. In: Proceedings of the workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, pp 57–64
43. Robbins H, Monro S (1951) A stochastic approximation method. In: The annals of mathematical statistics pp 400–407
44. Romero V, Toselli AH, Vidal E, et al (2019) Modern vs diplomatic transcripts for historical handwritten text recognition. In: Proceedings of the international workshop on pattern recognition for cultural heritage, pp 103–114
45. Sanchis-Trilles G, Ortiz-Martínez D, Civera J, et al (2008) Improving interactive machine translation via mouse actions. In: Proceedings of the conference on empirical methods in natural language processing, pp 485–494
46. Scherrer Y, Erjavec T (2013) Modernizing historical slovene words with character-based smt. In: Proceedings of the workshop on balto-slavic natural language processing, pp 58–62
47. Sen S, Hasanuzzaman M, Ekbal A, et al (2019) Take help from elder brother: Old to modern english nmt with phrase pair feedback. In: Proceedings of the international conference on computational linguistics and intelligent text processing, In press
48. Sennrich R, Haddow B, Birch A (2016) Neural machine translation of rare words with subword units. In: Proceedings of the annual meeting of the association for computational linguistics, pp 1715–1725
49. Snover M, Dorr B, Schwartz R, et al (2006) A study of translation edit rate with targeted human annotation. In: Proceedings of the association for machine translation in the Americas, pp 223–231
50. Stolcke A (2002) SRILM - an extensible language modeling toolkit. In: Proceedings of the international conference on spoken language processing, pp 257–286
51. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Proceedings of the Advances in Neural Information Processing Systems, pp 3104–3112
52. Szegedy C, Liu W, Jia Y, et al (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
53. Tang G, Cap F, Pettersson E, et al (2018) An evaluation of neural machine translation models on historical spelling normalization. In: Proceedings of the international conference on computational linguistics, pp 1320–1331
54. Tjong Kim Sang E, Bollmann M, Boschker R et al (2017) The CLIN27 shared task: translating historical text to contemporary language for improving automatic linguistic annotation. Comput Linguist Netherlands J 7:53–64
55. Tomás J, Casacuberta F (2006) Statistical phrase-based models for interactive computer-assisted translation. In: Proceedings of the international conference on computational linguistics/association for computational linguistics, pp 835–841
56. Torregrosa D, Forcada ML, Pérez-Ortiz JA (2014) An open-source web-based tool for resource-agnostic interactive translation prediction. Prague Bull Math Linguist 102:69–80
57. Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
58. Zens R, Och FJ, Ney H (2002) Phrase-based statistical machine translation. In: Proceedings of the annual German conference on advances in artificial intelligence, pp 18–32