Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 2901–2908
Marseille, 20-25 June 2022

Nunc profana tractemus. Detecting Code-Switching in a Large Corpus of 16th Century Letters

Martin Volk, Lukas Fischer, Patricia Scheurer, Bernard Schroffenegger, Raphael Schwitter, Phillip Ströbel, Benjamin Suter
Department of Computational Linguistics, University of Zurich
volk@cl.uzh.ch

Abstract

This paper is based on a collection of 16th century letters from and to the Zurich reformer Heinrich Bullinger. Around 12,000 letters of this exchange have been preserved, out of which 3100 have been professionally edited, and another 5500 are available as provisional transcriptions. We have investigated code-switching in these 8600 letters, first on the sentence-level and then on the word-level. In this paper we give an overview of the corpus and its language mix (mostly Early New High German and Latin, but also French, Greek, Italian and Hebrew). We report on our experiences with a popular language identifier and present our results when training an alternative identifier on a very small training corpus of only 150 sentences per language. We use the automatically labeled sentences in order to bootstrap a word-based language classifier which works with high accuracy. Our research around the corpus building and annotation involves automatic handwritten text recognition, text normalisation for ENH German, and machine translation from medieval Latin into modern German.

Keywords: language identification, historical languages, Latin, Early New High German

1. Introduction

“Nunc profana tractemus” (*Now let us deal with profane things*) writes Joachim Vadian in May 1548 in the middle of his letter to Heinrich Bullinger and switches from Latin to Early New High (ENH) German. In the first part of the letter he had written about the church and the bible in Latin, afterwards in German he deals with politics in France, Poland and Switzerland. Latin was the language of education and science all over Europe at the time. ENH German was used for every-day conversation which often resulted in mixed language exchanges (cf. (Jung, 2016) page 21).

Joachim Vadian was an intellectual from St. Gallen and in frequent letter contact with his colleague and friend Heinrich Bullinger in Zurich. The aforementioned letter is the 203rd letter in the preserved exchange between the two. They knew each other well. Did they always switch between languages when switching topics? Is there a pattern of code-switching for Vadian, or for others of the time?

We investigate the automatic detection of code-switching in this corpus of 16th century letters. How often do the writers switch between Latin and German? What other languages are used? How good are off-the-shelf language identifiers in distinguishing between medieval Latin and German when being trained on classical Latin and modern German texts?

2. Corpus Building

Heinrich Bullinger (1504-1575)[1] was a collaborator and successor of Huldrych Zwingli and an important multiplier for the ideas of the Reformation in Switzerland and Europe. From his extensive correspondence, some 2000 letters that Bullinger wrote and 10,000 letters that he received have been preserved. Most of the originals are kept in the Zurich State Archives and the Zurich Central Library. About 3100 letters have already been manually transcribed and professionally edited by the Swiss Reformation Studies Institute over the last three decades. Each letter comes with a German summary and scholarly footnotes. The edition has been published in printed form (Institut für Schweizerische Reformationsgeschichte, 1974 2019) and its PDFs can be searched online.[2] From the edition we have around 2850 letter texts in electronic form. In addition, another 5500 letters have been transcribed by various scholars and are also electronically available. The collection contributes to what has been termed the “Republic of Letters” by Hotson and Wallnig (2019) in order to honor the value of letter exchanges as a first-hand view to life and scholarship in the late middle-ages.

Our project aims at integrating all available knowledge sources including newly produced scan images for the 30,000+ manuscript pages into an online edition.[3] As part of this effort and for the sake of long-term and sustainable usage, we are compiling and annotating the text corpus.

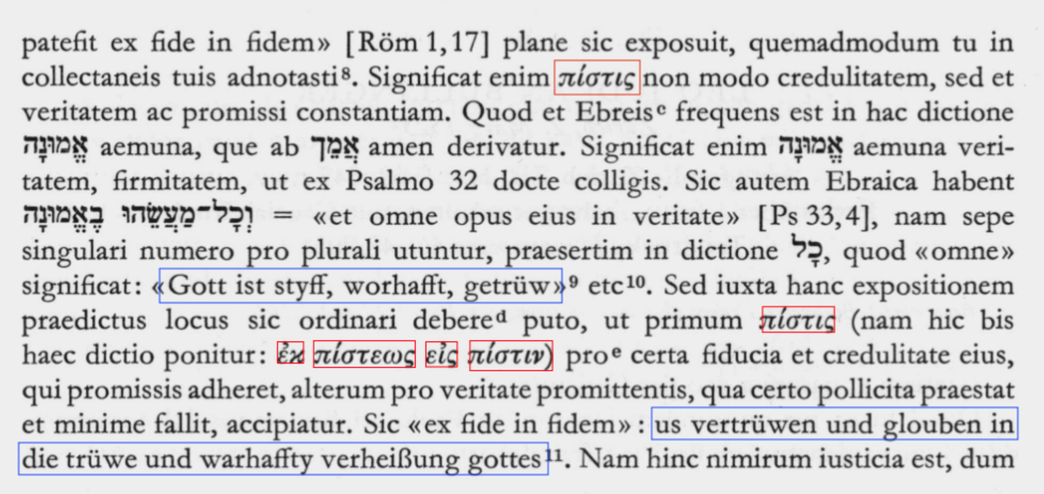
In order to build optimal handwritten text recognition (HTR) systems, we start with a scan-to-text alignment of the transcribed letters, which will allow users to appreciate the transcribed texts in sync with the scan images. On the basis of these scan-aligned transcriptions, we train systems for HTR to efficiently convert the remaining 3650 letters into electronic text.

In addition, we are building a text normalization sys-

[1] https://de.wikipedia.org/wiki/Heinrich_Bullinger

[2] http://teoirgsed.uzh.ch/

[3] Our version of the Bullinger letter exchange can be searched at https://www.bullinger-digital.ch



patefit ex fide in fidem» [Röm 1,17] plane sic exposuit, quemadmodum tu in collectaneis tuis adnotasti[8]. Significat enim πίστις non modo credulitatem, sed et veritatem ac promissi constantiam. Quod et Ebreis[c] frequens est in hac dictione אֱמוּנָה aemuna, que ab אָמֵן amen derivatur. Significat enim אֱמוּנָה aemuna veritatem, firmitatem, ut ex Psalmo 32 docte colligis. Sic autem Ebraica habent וְכָל־מַעֲשֵׂהוּ בֶּאֱמוּנָה = «et omne opus eius in veritate» [Ps 33,4], nam sepe singulari numero pro plurali utuntur, praesertim in dictione כָּל, quod «omne» significat: «Gott ist styff, worhafft, getrüw»[9] etc[10]. Sed iuxta hanc expositionem praedictus locus sic ordinari debere[d] puto, ut primum πίστις (nam hic bis haec dictio ponitur: ἐκ πίστεως εἰς πίστιν) pro[e] certa fiducia et credulitate eius, qui promissis adheret, alterum pro veritate promittentis, qua certo pollicita praestat et minime fallit, accipiatur. Sic «ex fide in fidem»: us vertrüwen und glouben in die trüwe und warhaffty verheißung gottes[11]. Nam hinc nimirum iusticia est, dum

Figure 1: Example letter from the edition (Leo Jud to Heinrich Bullinger on March 2nd 1525): The text is in Latin but comes with quotes in Hebrew, Greek (marked in red) and ENH German (blue). The superscript digits and characters are not in the original but refer to footnotes and comments in the edition.

tem for converting historical German words into modern German, and machine translation systems for translating medieval Latin into modern German (more in section 4).

We are in the process of turning the collection into a structured corpus in TEI XML format[4]. Each letter includes meta-information about sender, receiver, place and date, as well as a reference to current whereabouts of the original (or copy) in the form of a library signature.

The corpus takes caution to encode text-specific diacritical marks and special characters (e.g. e-caudata ę, combined small letters (like *o* over *u* in *ů*) or ligatures *æ*) as well as abbreviations (e.g. *key[serliche] m[ajestä]t* for *imperial majesty*).

The corpus texts are tagged with three levels of certainty.

1. The texts from the classical edition are the most reliable source.
2. The manually transcribed texts are of medium trustworthiness.
3. The texts that we automatically convert from scan images to electronic text via HTR are expected to have a character error rate of about 5%.

Based on the already transcribed letters we predict that our corpus eventually will have a total of around 5.5 million tokens. The texts deal with theological disputes but also with general news and everyday issues such as education, food and illness (cf. Beeler et al. (2018) for an overview and example letters in modern German).

For natural language processing, the Latin texts of the time have the advantage that the writing mostly follows the standard from Classical Latin, while ENH German comes with a great variety of spelling variants (e.g. *zyten, zytten, ziten, zitten, zeyten, zeiten* for modern German: *Zeiten*, English: *times*).

3. Code-Switching Detection

In order to prepare our corpus for subsequent processing steps like text normalisation for ENH German or machine translation for Latin we need to determine the language of each sentence and also check for possible code-switching within the sentences.

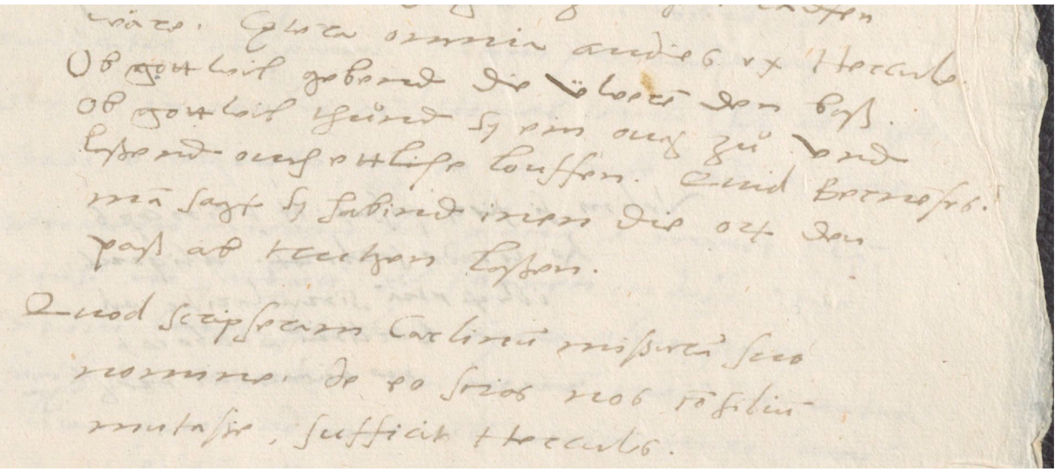
Many approaches for the detection of code-switching have been proposed. Since 2014 there has been a series of ACL workshops on Computational Approaches to Linguistic Code-Switching (including shared tasks) with the latest edition being the workshop in 2021[5]. Many contributions deal with code-switching in social media texts, but there are also papers on applications like named-entity recognition and machine translation of code-switched text. None of the papers deals with Latin-German code-switching.

It is striking that code-switching is discussed in the NLP community almost exclusively for current languages. Even a recent, comprehensive survey of the social and linguistic aspects of code-switching (Doğruöz et al., 2021) does not mention code-switching for historical languages. Our paper covers new ground in this area.

Let's look at the few publications about code-switching in historical texts. Garrette et al. (2015) deal with 16th century texts that mix English, Spanish and Nahuatl in 349 early American writings. The paper presents a new language model for the OCR system Ocular that is designed to handle characteristics of printed historical

[4] https://tei-c.org/

[5] `https://aclanthology.org/volumes/2021.calcs-1/`



Caetera omnia audies ex Hercule.
Ob gott wil gebend die üweren den baß.
ob gott wil thůnd sy ein oug zů und
lassend ouch ettliche louffen. Quid Bernenses?
man sagt sy habind inen die ort den
paß ab trutzen laßen.
Quod scripseram Carlinum missurum suo
nomine, de eo scias nos consilium
mutasse, sufficit Hercules.

Figure 2: Excerpt from a letter by Johannes Fabricius to Heinrich Bullinger (June 26, 1562): The text and the handwriting switches between **Latin** and ENH German: **Caetera omnia audies ex Hercule.** *Ob gott wil, gebend die üweren den baß; ob gott wil, thůnd sy ein oug zů und lassend ouch ettliche louffen.* **Quid Bernenses?** *Man sagt, sy habind inen die Ort den paß abtrutzen laßen.* **Quod scripseram Carlinum missurum suo nomine, de eo scias nos consilium mutasse; sufficit Hercules.**

documents: code-switching and orthographic variability. They evaluated their new approach on five books by different printers, typefaces and authors and found a clear improvement when including the code-switch model. This could be interesting for us when we exploit the different hand-writing systems for Latin and ENH German in our letter collection.

Schulz and Keller (2016) discuss intra-sentential code-switching in medieval sermons with mixed Latin and Middle English. Their goal is to identify code-switching within noun phrases. Their small corpus consists of 159 sentences where they classify each token for language and PoS tag. Training a CRF classifier shows clear improvements over the baseline.

Liu and Smith (2020) investigate code-switching into Latin, English, French, and Greek in a corpus of 1406 German books from the 17th to 19th centuries. They use a language detection service in the internet[6]. This service claims to support 165 languages. We tested their online demo with 25 strings of 20 characters (sentence-initial) from German and Latin truncated sentences from our corpus. We found this language identifier to be correct for 18 of 25 truncated Latin sentences and 11 of 25 truncated German sentences (with another 6 sentences where German ranked among the top 3 languages). None of the German sentences was labeled as Latin and vice versa.

Since we are unable to restrict the search in this online service to the two languages in question, shorter strings sometimes go astray (e.g. the German greeting *Gott mitt üch* (EN: *God with you*) is labeled as Swedish, Icelandic or English, and the Latin farewell *Hallerus tuus* (EN: *yours Haller*) is recognized as Estonian or Polish). The goal of Liu and Smith (2020) is to predict code-switching in order to prepare their OCR systems for embedded languages.

We will first report on our experiences with another off-the-shelf language identifier and then introduce our own approach.

3.1. Sentence-based Language Identification

There are various systems for automatic language identification (for a recent detailed overview see (Jauhiainen et al., 2019)). Most of them work by classifying letter-n-grams. One of the best-known systems is `langid.py` (Lui et al., 2014) which comes with a pretrained model for 90+ languages including German and Latin which are of interest here. In our experience this language identifier usually works reliable for input strings of length 40 characters and more (cf. (Volk and Clematide, 2014) where we introduce a method to detect intra-sentential code-switching between English, French, and German in a corpus of alpine texts from the last 150 years).

So, our hope was that `langid` would distinguish fairly between German and Latin sentences in our corpus. Therefore we tested `langid` with its pretrained model and with the restriction to choose only between German and Latin.

We soon realized that `langid` has a clear preference for German over Latin. About 2% of the Latin sentences in our corpus were incorrectly labeled as German (and hardly any German sentence was erroneously classified as Latin). For example: *Non habet facultates amplas, nec frater meus habet.* with 51 characters was misclassified as German.

We suspected this skew may be because `langid` was trained on classical Latin and modern German (rather than on medieval Latin and historical German). There-

[6] `https://detectlanguage.com`

Latin to **German**	Pater in morbo semel et iterum clamavit: **“Louff, Hans, du findst mich sunst nitt mee!”**
German to **Latin**	Ich hette wol mit sant Thoma mogen reden: **“Domine, quo vis, mittas me, praeter ad Indos!”**

Table 1: Examples of code-switches for direct speech within quotation marks

fore we tested langid against Caesar’s famous De Bello Gallico which certainly does not contain anything but Latin. The first book consists of 314 sentences in 54 sections. The average sentence length is 183 characters, and only 20 sentences are shorter than 50 characters.

Out of the total of 314 sentences langid classified only 272 sentences (86.6%) as Latin. The other 42 sentences were classified into 10 different languages with - unsurprisingly - labels in other Romance languages being the most frequent (13 Italian, 9 Portuguese, 7 Spanish, 4 French), and no German.

In order to assess the performance of langid on short strings, we artificially cut all sentences down to the first 50 characters. Then only 177 sentences are classified as Latin (56.4%) while the remaining labels are distributed over 17 other languages. And if we shorten all sentences to 20 characters, then only 26 sentences with Latin labels are left (8.3%) while the other sentences are distributed over 26 other languages with English (!) being at the top with 50 sentences. This experiment clearly shows that langid is unreliable for detecting Latin sentences, and in particular, that it is unusable for sentences that are shorter than 50 characters.

If we restrict langid’s search to Latin and German, then it misclassifies 2.2% of the truncated 50-character Bello Gallico sentences and 26.7% of the 20-character sentences as German. This shows that langid has a bias towards German even for Classical Latin.

We also realized that langid is not reliable for identifying the few French and Italian letters in our corpus. We therefore searched for French and Italian letters with frequent function words *je, le, l’, che, ...*. We identified seven French letters and two Italian letters and removed them from our further experiments. We did the same for Greek (based on Greek characters) and omitted two letters from our experiments that are completely in Greek. This, of course, leaves inter- and intra-sentential code-switching in German and Latin letters with Greek, Hebrew, French and Italian in our corpus. Among them Greek is by far the most frequent and occurs in around 1100 German or Latin sentences. In contrast, we found Hebrew, French and Italian only in some dozen sentences. Figure 1 gives an extreme example letter from the printed edition with five languages. Figure 2 shows the scan of an original letter with Latin and German.

We decided to train our own language identifier based on an implementation by our colleagues Lenz Furrer and Samuel Läubli. The character 3-gram system loosely follows the ideas put forth in chapter 4 of (Jurafsky and Martin, 2018). We manually selected 150 sentences in Latin and the same number of sentences in ENH German from our corpus (with roughly 2500 tokens each) as training material. Training on this limited data set resulted in a much more reliable classification. When we tested our German-Latin classifier against the Bello Gallico test corpus, we found a 100% accuracy down to the truncated 20-character sentences. When we cut further to 10-character strings, then we observe a small 2.2% error rate. This clearly indicates the superior performance of our classifier (which we termed FurL based on the authors’ names).

With FurL we classified around 165,500 sentences of our corpus as Latin and 39,600 sentences as German. We automatically classified a letter in our corpus as a code-switching document by comparing the number of characters in sentences that we had labeled as ENH German to those labeled as Latin. If the number of characters exceeded 3% for either language, then we counted the letter as code-switching letter. This comparison worked well for short letters but is too coarse for long letters. Therefore we also counted letters as code-switching if they have at least two sentences with at least 30 characters in the other language. The combination of these criteria resulted in the frequencies listed in the following table for the letters in our corpus.

Code-sw ENHG	Code-sw Latin	ENHG	Latin
688	1330	920	5309
2018		6229	

On the left we list the number of letters in our corpus with code-switching (based on sentence language labels); on the right we list the monolingual letters (i.e. without code-switching).

For example: “Code-sw ENHG” stands for letters in ENH German that contain sentences which we automatically labeled as Latin. So, we find that a total of 2018 letters (24.5% out of a total of 8247 letters) contain code-switches on the sentence level.

Manual inspection quickly showed that our corpus contains many sentences with intra-sentential code-switching. In other words, the author changes between German and Latin within the same sentence. We first hypothesized that such code-switching would be signalled by quotation marks.

We therefore extended our experiment so that the language identifier determined the language of the string within a pair of quotation marks (if the string is longer than 8 characters which seems like a reasonable lower bound). If that language differed from the language of the matrix sentence surrounding it, then we count this as intra-sentential code-switching. We found that 117

Latin to **German**	Crastino comitia erunt Domus tantum Dei propter dissidium Zuziensium et Samadensium **von stok und galgen wegen**.
German to **Latin**	Dann **Galli nostri** treüwend unnd erschreckend mengem das hertz, das er hinschlichen last, **ne privetur stipendio**.

Table 2: Examples of code-switches within a sentence

sentences that we had labeled as ENH German contained Latin text in quotation marks. And we found 106 sentences in the opposite direction. See table 1 for example sentences.

Upon closer inspection of the corpus it became clear that intra-sentential code-switching is not limited to phrases in quotation marks. We therefore decided to invest into word-level language identification.

3.2. Bootstrapping Word-based Language Identification

In order to do lexicon-based language identification for each word in a corpus one needs large word lists. We could possibly get such word lists for medieval Latin and ENH German from other corpora of this historic period. But this will always raise the issue of the lexical coverage with respect to our corpus. Therefore we developed a bootstrapping approach using our own corpus to build the disjoint vocabulary lists needed for the two languages in question. Our approach has the advantage that it is easily applicable to similar corpora and does not require external resources.

Step 1: We used the sentences in our corpus that were labeled by both language identifiers (`langid` and FurL) in the same way (either both identifiers say Latin or both say German.) We ignored all tokens that contained digits and removed all punctuation symbols from the beginning and the end of the tokens. We also removed square brackets from within the tokens (e.g. *Th[obias]* → *Thobias*). We preserved upper and lower casing, suspecting that it might be useful in particular for German. In this way we collected the German and Latin vocabularies from all letters in our corpus. This resulted in 158,663 types (= unique words) for Latin and 72,089 types for German. Both these type lists contain words from the other language since we rely only on the language label of the complete sentence. But many sentences have code-switching parts - as we will show.

Step 2: We filtered the collected vocabularies. We kept all types in Latin that were not in the German list. And in addition we kept all types in Latin where the Latin occurrence frequency is at least 10 times the frequency in the German list. This is meant as a work-around for the noise that we inadvertently gather in step 1. See examples in table 4.

We did the same for German except that we kept types that occurred at least 5 times as often as in the Latin list. This difference (factor 10 vs. factor 5) is meant to counteract the skew in the overall token numbers.

After filtering we have disjoint vocabularies of the following sizes. For Latin we keep 152,333 types and for German 64,652 types. The reasons for the high type counts are the morphological richness of both Latin and German, frequent person and geographical names, and in the case of German the spelling variations that are due to a lack of orthographic standardization at the time.[7] For example, the Latin word *dominus* has the forms *dominę, domino, dominum, domini, dominis, dominorum* and appears also with the attached conjunction *-que* as *dominusque, dominumque, dominorumque, ...* In total, the Latin vocabulary contains more than 5400 types that end in *-que*.

In addition, the upper-lower case distinction adds to the count although a conversion to all lower case only reduces the type count by 8.5% for Latin and by 5.4% for ENH German. Interestingly, compounding is not a factor as it would be for modern German. ENH German writers did not glue words together (with few exceptions in our corpus: *bluottfrünntschafften, kouffmansguetteren, schuuolgeschäfften*, in English: *blood friendship, trading goods, school business*).

Step 3: We apply the German and Latin disjoint vocabularies for word-based language classification. For each token in the letter texts we determine whether it is in the German list or in the Latin list or unknown. We apply the same tokenization rules as in step 1. A token gets the label 'unknown' if it contains a digit or if it was excluded from the vocabularies because of the overlap between the two languages. We also exclude tokens that consist only of one character (e.g. the abbreviation *d.* for *dominus*), since they do not help language identification.

Step 4: In all sentences that were automatically labeled as German we search for at least two subsequent tokens labeled as Latin. We count these as code-switching sentences. We identify the code-switched token span by including "embedded" unknown tokens into the span. For example, after word-level language classification we obtain the following annotation

At... praecipuum, quod erat, eius oblitus eram, et <u>alter</u> **consul dixit**, *ir söllind umb üwer schuld khein sorg mee han, und ist ...*

All words in bold face have been recognized as Latin, all words in the end of the sentence as German. The word <u>alter</u> cannot be classified by itself since it occurs both in the Latin and German vocabularies (meaning

[7] In addition there are occasional typos introduced by the transcribers. For example for modern day *Kreuzlingen* (a town on lake Constance) we find the spelling variations *Creutzlingen, Crutzlingen, Crützlingen*, but also the erroneous *Creutzilngen*.

Latin to **German**	Sed, ut dixi, nulla seditio timenda, cum omnia quieta, et si res ad summum veniet, nostri cum episcopo et ipso **Gottshuß** iure experiri statuerunt.
German to **Latin**	Hatts zů Gennt und Brugk publiciert, darnach gen Antorff kon; nieman aber hatt **simpliciter** drin bewilligen wellen dann die zů Löven.

Table 3: Examples of single word switches (names and loan words) within a sentence

token	freq(DE)	freq(LA)	vocab
Albrecht	41	1	German
Alexander	9	18	undec
Africa	2	10	undec
Augustinus	5	147	Latin
in	9298	50,340	undec
bis	259 *up to*	145 *twice*	undec
breve	9 *letter*	67 *short*	undec
briefen	22 *letters*	1 –	German
dies	17 *this*	1236 *day*	Latin

Table 4: Examples from the overlapping vocabulary German-Latin with their frequencies and the language decision after step 2 (column 4).

other in Latin vs. *old* in German). Since the two surrounding words are both in Latin, we label *alter* as Latin in this context and include it in the Latin text span.

If the first word in a sentence has been classified as "unknown", we use the language of the following word for the disambiguation. In analogy, if the last word in a sentence is "unknown", we copy the label of the preceding word. This leaves "unknown" words with unequal contexts (i.e. one context word in German and one in Latin) open for classification. If such "unknown" words are preceded by a comma or an opening parenthesis, they get the language of the following word. And accordingly, if they are followed by a comma or a closing parenthesis, we assign the language of the preceding word. The few remaining words are left as undecidable.

Results: In this way we identify 1505 sentences with intra-sentential code-switching. We also searched in the opposite direction: For all sentences labeled as Latin, we searched for two or more subsequent tokens labeled as German. This results in 1563 sentences with internal code-switching. See table 2 for example sentences.

In order to assess the quality of our word-level language identification we performed a manual evaluation of 50 random sentences where we had automatically determined code-switching. We skipped 'easy' sentences where the language change was indicated by quotation marks, parentheses or a semicolon (all of which could be regarded as a language boundary or a code-switching trigger). In the evaluation we focused on 'hard' sentences where the language changed without any trigger (as in table 2). Our evaluation revealed that we correctly label more than 99% of the tokens as Latin and German. The 50 sentences add up to 1075 tokens (532 labeled as German, 429 labeled as Latin and 114 unknown). After the disambiguation of the unknown tokens as either German or Latin based on the algorithm in step 4, we are left with only 11 undecidable and erroneously labeled words.

We realize that our approach favours precision over recall. It disregards cases where only a single word occurs in the other language in order to boost the reliability of the decision. We could include single-word occurrences if the word is "prominent" because of being above a certain frequency threshold or a length threshold. We experimented with a length threshold of more than 8 characters and found that many hits are Latin loan words in German sentences, or German person and location names in Latin sentences (see table 3 for examples).

3.3. Word-internal Language Mix

In general, it is difficult to clearly determine switches between German and Latin within words. But in cases of special characters it is still possible. For example, the umlauts *äöü*, the German-specific character *ß*, still in use today, and the historical compound characters *ů*, *uͤ*, *oͤ* do not occur in Latin and are therefore good indicators for German. However, in our corpus we find words with these characters that exhibit Latin suffixes: *Küngßfeldiana, Merßburgensis, Straßburgo*. On the other hand, medieval Latin uses the special character *ę* (e-caudata), but we find some German words in our corpus that also use this character: *moęglich, prędigen, ußerlesenę* (in English: *possible, to preach, selected*).

4. Corpus Processing

Precise language labels for sentences and sentence-internal code-switch sequences are a prerequisite to all further processing of the letters in our corpus.

4.1. Handwritten Text Recognition

Automatic HTR has seen substantial improvements in recent years due to neural network learning. This has been demonstrated by the European READ project, which has led to the development of the Transkribus platform.[8] The READ project shows that already 10 pages of manually corrected ground truth will lead to

[8] `https://readcoop.eu/`

good recognition rates for historical handwritten documents. Depending on the regularity of the handwriting, a Character Error Rate of around 5% can be expected. Our experiments with the Transkribus platform (cf. Mühlberger et al. (2019)) indicate that this goal is realistic.

The 3,100 high-quality edited letters mentioned above stem from about 300 writers. This includes 11 writers with more than 50 transcribed letters. When we add the 5,500 letters of scholarly transcriptions, we get 13 writers for whom we have 100 and more transcribed letters. This is great data for training the HTR system.

In order to do so, we need to distinguish between German and Latin letters (not least because the handwriting of some authors changes when they switch from German to Latin, as can be seen in figure 2 above). We automatically align the already transcribed texts to the scans, which will provide us with a ground truth of about 8,000 letters. These are good conditions for training different HTR models under different circumstances. For example, the training set can be adapted so that it contains not only a different number of pages, but also a different distribution of authors, which is heavily skewed.

The several hundred writers for whom we have only a few letters and few or no transcriptions are one of the challenges of the current project. It does not make sense to train specific HTR models for the rare writers. Instead, we develop methods to deal with these sparse-data cases. Our goal is to find the HTR model that performs best for each of these writers. One way is to build classifiers that group the handwritings based on similarity characteristics. Another approach is to have the letters of these non-frequent writers analysed by all of our HTR models and to automatically evaluate the output (i.e., the automatically recognised text) against word lists (which we derive from the transcribed letters) and statistical language models.

4.2. Text Normalization for ENH German

In order to make the letters from our corpus accessible and searchable we built a system for normalizing the medieval German words into modern German (stored as an additional layer of information in addition to the original). Our fine-grained code-switching detection is an important prerequisite to applying the normalization not only to ENH German sentences but also to parts of Latin sentences in ENH German.

Our normalization system uses a standard transformer model and thus provides context-dependent normalization. We frame the task as a spelling correction problem and assume that ENH German texts are merely modern German texts with spelling errors. Therefore, instead of manually annotating ENH German samples with their expected normalization in order to create a training corpus, we take a monolingual corpus of modern German and synthetically generate a source-side corpus by introducing spelling corruptions.

To create these corruptions, we defined a list of possible string substitutions, such as replacing certain characters with phonologically similar characters (e.g. t - d), or replacing a single character with its duplication and vice versa (e.g. t - tt). These predefined corruptions are then randomly applied to individual modern German training samples. The final parallel training corpus does not contain any real historical ENH German data.

As corpus of modern German we used the sub-corpus of the *Deutsches Textarchiv* from the period 1900-1999[9]. To this we appended a small project-internal collection of modern German translations and summaries of Bullinger letters as in-domain data.

On a test set of 1201 tokens, the normalization model achieves a word error rate of 14.2%, which is comparable in quality to other state-of-the-art models for historical text normalization (Makarov and Clematide, 2020).

4.3. Machine Translation from Medieval Latin to Modern German

As most letters in the Bullinger correspondence are written in Latin, we will provide a German translation, which will be automatically generated by a customised machine translation system. Therefore, one of the project's milestones is the development of an MT model that is optimized for Medieval Latin (much like (Martínez Garcia and García Tejedor, 2020) for Latin to Spanish). Our objective is to beat the quality of Google Translate for this specific task.

We collected translated texts in Latin and German (and English) as training material from various online sources ranging from classical Latin texts to modern day Vatican publications. We are investigating which of these diverging sources are most useful for building the best MT system for medieval Latin (cf. (Fischer et al., 2022)). We plan to release this parallel corpus, too, as a resource for other researchers.

It is clear that intra-sentential code-switching is a challenge for MT. One simple option is to combine text normalization for ENH German to modern German and MT for Latin to modern German. We will test also the more elaborate approach by Gupta et al. (2021).

5. Conclusion

This paper introduces a large corpus of 16th century letters in Latin and Early New High German for studies in history, linguistics, and theology. The corpus also contains a few letters in French, Greek and Italian. However, in this paper we focused on the detection of code-switching on the sentence and on the word level between Latin and ENH German. We found that the pretrained model of the language identifier `langid` does not reliably distinguish between Latin and German. Therefore we trained our own classifier. We showed that this special-purpose language identifier trained on

[9] https://www.deutschestextarchiv.de/download

only 150 sentences works well for binary German vs. Latin sentence-level language labeling.

Based on this sentence classification, we bootstrapped a word-level language identifier which works with very high accuracy and reliably finds sentence-internal code-switches. Our method is easily applicable and guarantees high lexical coverage which is particularly important for languages like ENH German with many spelling variants.

Our corpus integrates various text sources (edited, transcribed and automatically converted). We will make both the corpus and the digital edition available online within the next year. We predict that the corpus eventually will size up to roughly 4 million tokens in Latin and 1.5 million in ENH German.

Acknowledgments

We gratefully acknowledge project funding provided by various sponsors through the UZH Foundation (see `www.bullinger-digital.ch/about`).

6. Bibliographical References

Luca Beeler, et al., editors. (2018). *Nüwe Zyttungen. Der Briefwechsel des Reformators Heinrich Bullinger*. Scheidegger & Spiess.

Doğruöz, A. S., Sitaram, S., Bullock, B. E., and Toribio, A. J. (2021). A survey of code-switching: Linguistic and social perspectives for language technologies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 1654–1666.

Fischer, L., Scheurer, P., Schwitter, R., and Volk, M. (2022). Machine translation of 16th century letters from Latin to German. In *Proceedings of 2nd Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA) at LREC-2022*, Marseille.

Garrette, D., Alpert-Abrams, H., Berg-Kirkpatrick, T., and Klein, D. (2015). Unsupervised code-switching for multilingual historical document transcription. In *Proceedings of Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, page 1036–1041, Denver.

Gupta, A., Vavre, A., and Sarawagi, S. (2021). Training data augmentation for code-mixed translation. In *Proceedings of the 2021 Conference of the North American Chapter of the ACL: Human Language Technologies*, page 5760–5766.

Howard Hotson et al., editors. (2019). *Reassembling the Republic of Letters in the Digital Age. Standards, Systems, Scholarship*. Göttingen University Press.

Institut für Schweizerische Reformationsgeschichte, editor. (1974-2019). *Heinrich Bullinger Briefwechsel. Briefe von 1523 bis 1547*, volume 1-19. Theologischer Verlag Zürich.

Jauhiainen, T., Lui, M., Zampieri, M., Baldwin, T., and Lindén, K. (2019). Automatic language identification in texts: A survey. *Journal of Artificial Intelligence Research*, 65:675–782.

Jung, M. H. (2016). *Die Reformation. Wittenberg - Zürich - Genf. 1517-1555.* Marix-Verlag.

Jurafsky, D. and Martin, J. H. (2018). *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, Upper Saddle River, NJ, 3 edition.

Liu, S. and Smith, D. (2020). Detecting *de minimis* code-switching in historical German books. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1808–1814, Barcelona.

Lui, M., Lau, J. H., and Baldwin, T. (2014). Automatic detection and language identification of multilingual documents. *Transactions of the Association for Computational Linguistics*, pages 27–40.

Makarov, P. and Clematide, S. (2020). Semi-supervised contextual historical text normalization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Martínez Garcia, E. and García Tejedor, A. (2020). Latin-Spanish neural machine translation: from the Bible to Saint Augustine. In *Proceedings of 1st Workshop on Language Technologies for Historical and Ancient Languages at LREC*, pages 94–99, Marseille.

Mühlberger, G., Seaward, L., Terras, M., and 51 more authors. (2019). Transforming scholarship in the archives through handwritten text recognition: Transkribus as a case study. *Journal of Documentation*, 75(5):954–976.

Schulz, S. and Keller, M. (2016). Code-switching ubique est - language identification and part-of-speech tagging for historical mixed text. In *Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 43–51, Berlin. Association for Computational Linguistics.

Volk, M. and Clematide, S. (2014). Detecting code-switching in a multilingual alpine heritage corpus. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 24–33, Doha, Qatar.