Detecting *de minimis* Code-Switching in Historical German Books

Shijia Liu **David Smith**
Khoury College of Computer Sciences
Northeastern University
liu.shij@northeastern.edu, dasmith@ccs.neu.edu

Abstract

Code-switching has long interested linguists, with computational work in particular focusing on speech and social media data (Sitaram et al., 2019). This paper contrasts these informal instances of code-switching to its appearance in more formal registers, by examining the mixture of languages in the Deutsches Textarchiv (DTA), a corpus of 1406 primarily German books from the 17th to 19th centuries. We automatically annotate and manually inspect spans of six embedded languages (Latin, French, English, Italian, Spanish, and Greek) in the corpus. We quantitatively analyze the differences between code-switching patterns in these books and those in more typically studied speech and social media corpora. Furthermore, we address the practical task of predicting code-switching *from features of the matrix language alone* in the DTA corpus. Such classifiers can help reduce errors when optical character recognition or speech transcription is applied to a large corpus with rare embedded languages.

1 Introduction

Code-switching, the linguistic phenomenon where speakers or writers alternate between different languages in a single utterance or statement, is commonly seen in bilingual or multilingual communities. In the last few decades, code-switching has drawn scholarly attention in computational linguistics and natural language processing from many different perspectives (Sitaram et al., 2019). Researchers from formal linguistics, psycholinguistics, sociolinguistics, philosophy, anthropology, and elsewhere have considered the phenomenon (Nilep, 2006). In formal linguistics, interest has risen in studying syntactic and morphosyntactic constraints on language alternation (Nilep, 2006). In sociolinguistics, there has been a focus on the social context in conversations where code-switching happens. For example, Blom and Gumperz (1972) proposes the dichotomy of *situational* and *metaphorical* code-switching, which serve as indicators of whether different languages or language varieties are used in different social situations.

In natural language processing, there is extensive work studying code-switching from an engineering perspective. Different NLP tasks have been proposed on code-switching corpora, such as language ID (Solorio et al., 2014; Sequiera et al., 2015), named entity recognition (Aguilar et al., 2018; Singh et al., 2018) , POS tagging (Solorio and Liu, 2008; Vyas et al., 2014), sentiment analysis (Vilares et al., 2015), automatic speech recognition (Chan et al., 2014; Weiner et al., 2012), question answering (Chandu et al., 2018), etc. Indeed, with the abundance of code-switched speech and (informal) text data, there are many choices of NLP directions that one can pursue.

Code-switching, as a language phenomenon, is usually considered informal. It is often found in speech and in casual text, such as social media (Sitaram et al., 2019); however, code-switching also appears in formal settings, such as newspaper reports, or in this paper, books. Table 1 shows examples from the Deutsches Textarchiv (DTA) corpus of code-switching from German into Latin, English, French, and Greek. Based on such observations, we ask: from a quantitative perspective, precisely how different is formal, "scholarly" code-switching from its informal usage? Furthermore, can we predict which books

Proceedings of the 28th International Conference on Computational Linguistics, pages 1808–1814
Barcelona, Spain (Online), December 8-13, 2020

Embedded language	Code-switched text
Latin	. . . die *fortitudo animi, magnitudo* (c. 5.) seinem Sohne schildert, . . .
English	. . . die man das Westende nennt, *the west end of the town*, und wo die vornehmere und minder beschäftigte Welt lebt.
French	. . . eine frivole Laune, ein "*car tel est notre plaisir*" des Geistes . . .
Greek	Diess ist das Spinnrad des βάθος; diess ist das Spinnrad, welches die Gedanken spinnet, . . .

Table 1: Code switching examples in the Deutsches Textarchiv, embedded language spans in red italics.

will host code-switching, so as to reduce language-coding and transcription errors in mass digitization? If we can, how would these prediction tasks be different from common NLP tasks in informal code-switching settings?

To measure the difference between formal and informal code-switching, we evaluate metrics for characterizing code-switching across corpora. Specifically, we are interested in: 1) how "unequal" the distribution of different languages is in a corpus; 2) how frequently the switching occurs; and 3) whether the switching happens in a periodic or aperiodic manner. We want to obtain metrics to answer the questions above quantitatively, and we employ these metrics to get a sense of how code-switching characteristics differ among corpora.

Besides this descriptive task, we are interested in practical tasks for predicting code-switching. There has been previous work formalizing code-switching detection in historical texts as a language ID task (Schulz and Keller, 2016; Sprugnoli et al., 2017), and models such as Conditional Random Fields (CRF) have been deployed to classify words as in one language or another. However, such approaches fail to work in the following scenario: when large collections of page images are transcribed with optical character recognition (OCR) or when large audio collections are transcribed by speech recognition, we do not always know *a priori* which languages will be included. Including hypotheses from multiple languages in a transcription model can reduce accuracy in the majority matrix language. Including a model trained only on the matrix language results in near-zero accuracy in transcribing embedded languages. When the page containing the Greek example from Table 1 is run through a German-only OCR model, the output contains *BxIos* in place of βάθος. We therefore experiment with two predictive tasks. First, at the level of the book, can we predict the presence of code-switching *using only features of the matrix language text*? Second, working sequentially through a text, can we predict when code-switching will occur?

The rest of the paper is organized as follows: §2 gives an introduction on the metrics we use to characterize code-switching. §3 describes the datasets we analyze and the results of the metrics described in §2. §4 shows the results for the two predictive tasks described above applied to historical books from the DTA. Finally, §5 summarizes our conclusions and outlines possible future directions.

2 Metrics for Characterizing Code-switched Corpora

In this section we briefly introduce three metrics for measuring characteristics of code-switched corpora. The metrics, as defined by Guzmán et al. (2017) on the basis of previous work, include: **M-index**, measuring the inequality of the distribution of languages; (normalized) **I-index**, measuring the frequency of switching in a code-switched corpus; and **burstiness**, measuring the degree of periodicity in code-switching patterns in the corpus.

The Multilingual Index (M-index), developed by Barnett et al. (2000), is a token-count-based measure that "quantifies the inequality of the distribution of language tags in a corpus of at least two languages" (Guzmán et al., 2017). It is defined as follows:

$$M \equiv \frac{1 - \sum {p_j}^2}{(k-1)\sum {p_j}^2} \tag{1}$$

where k denotes the total number of languages in the corpus, and p_j is the proportion of the number of

words in language j in the corpus. M-index ranges from 0 (indicating a completely monolingual corpus) to 1 (denoting each language in the corpus has the same number of words).

The Integration Index (I-index) measures the frequency of code-switching. It "describes the probability of switching within a text" (Guzmán et al., 2017). The unnormalized I-index is calculated as:

$$I \equiv \frac{1}{n-1} \sum_{1 \leq i=j-1 \leq n-1} S(l_i, l_j) \tag{2}$$

where n is the total number of tokens in the text, l_i denotes the language of token i, and $S(l_i, l_j) = 1$ if $l_i \neq l_j$ and 0 otherwise. As we can see, this quantity measures the proportion of number of switch points relative to the total number of tokens in the corpus.

The unnormalized I-index, however, does not consider the underlying language distribution in the corpus. For example, the unnormalized I-index would not be close to 1 unless the M-index of the corpus is close to 1, indicating that each language in the corpus is equally distributed. In order to decouple this metric from the underlying language distribution of the corpus, a normalized version of the I-index is developed by Bullock et al. (2019) and is computed as follows:

$$I_{normalized} \equiv \frac{I - L}{H - L} \tag{3}$$

where H and L are the upper and lower bounds of the unnormalized I-index, respectively. Let n be the total number of tokens in the text, k be the total number of languages, and n_i be the number of tokens in language i. We can then define:

$$L \equiv \frac{k-1}{n-1} \tag{4}$$

$$H \equiv \min\left(\frac{2 \cdot (n - \max n_i)}{n-1}, 1\right) \tag{5}$$

Note that $I_{normalized}$ ranges from 0 to 1, representing the absolute minimum and maximum numbers of possible switches within the corpus, regardless of the underlying language distribution. Therefore, one can direct compare this metric across different corpora.

Burstiness (Goh and Barabasi, 2008) "quantifies whether switching occurs in bursts or has a more periodical manner" (Guzmán et al., 2017). It is defined as:

$$Burstiness \equiv \frac{\sigma_\tau - m_\tau}{\sigma_\tau + m_\tau} \tag{6}$$

where σ_τ and m_τ denote the standard deviation and the mean of the language spans, respectively. Burstiness ranges from -1 (periodic code-switching in corpora) to 1 (aperiodic, less predictable code-switching in corpora).

3 Datasets and Analysis

We now describe our corpora and analyze them for their patterns of code-switching.

3.1 Corpus Descriptions

In this paper, we focus on the Deustches Textarchiv (DTA) [1] corpus, which contains manual transcriptions of 1,406 historical German books from the 17th to the 19th centuries. The corpus contains 131,679,459 tokens in total. Until about the 1930s, German was usually written in a "blackletter" font named "Fraktur", but other languages in the Roman alphabet were written in a Roman font, called "Antiqua" in German. Since the DTA encodes this font information, we are able to identify text written in Roman-script languages other than German. We then use an off-the-shelf language identification API [2]

[1] `http://www.deutschestextarchiv.de`: Since the DTA is released under a CreativeCommons Non-Commercial License, we are preparing a public release of our annotations.

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

Corpus	DTA	LinCE (SPA-ENG)	(NEP-ENG)	(HIN-ENG)	(MSA-EA)
M-index	0.0008	0.9633	0.9668	0.6243	0.4396
Normalized I-index	0.6827	0.0489	0.1449	0.0637	0.0051
Burstiness	0.2320	-0.0709	-0.0780	-0.0068	-0.0967

Table 2: Metrics reveal the divergence of the Deutsches Textarchiv from informal corpora.

to label the "Antiqua" text spans with their corresponding languages. To eliminate errors made by the API, we then perform manual correction on all the labelled spans. We easily identify spans of embedded Greek by locating Greek UTF-8 characters.

For comparison, we also use the LinCE corpora (Aguilar et al., 2020) to characterize differences between formal and informal code-switched text. LinCE combines Twitter and Facebook data from ten corpora, and the language in these corpora is more informal. The corpora cover four different code-switched language pairs: Spanish-English, Nepali-English, Hindi-English, and Modern Standard Arabic-Egyptian Arabic. Overall, LinCE contains 64,326 posts with 953,813 tokens. Although we realize that the two corpora differ in many other dimensions, such as cultural context, topic, text length, language pairs and so on, we believe that formality is a crucial factor that needs to be taken into account for our comparison.

3.2 Corpus Comparison

Results for the code-switching metrics introduced in §2 are shown in Table 2. We can see that the M-index for the DTA corpus is very close to zero, while the numbers for the LinCE corpora are significantly greater than zero, suggesting that the language distribution is much more skewed in the DTA corpus compared to the LinCE corpus. We also observe that both the normalized I-index and the burstiness of the DTA corpus are much greater than those of the LinCE corpora, indicating that the code-switching phenomenon is more frequent (regardless of the underneath language distribution) and less periodical in the DTA corpus than in the LinCE corpora. Furthermore, a high normalized I-index of the DTA corpus implies that the non-German blocks in the corpus are quite short so that the probability of switching back to German in those non-German tokens is high, while in LinCE corpora passages in a particular language are usually longer.

4 Predictive Tasks for Studying Code-Switching

Results in §3.2 demonstrate that the code-switching patterns in historical books are significantly different than in usually studied domains, such as social media. In this section, we consider two tasks that are uniquely suitable for investigating code-switching in historical books. As discussed in §1, both tasks aim to improve OCR performance on the books. For both tasks, we utilize a 80%/10%/10% train/dev/test book-level split of the DTA corpus.

For the first task, we predict whether there are non-German languages present in a book for the DTA corpus. For our book-level baseline, we pick the top 1,000 words in each book and use word count as features. Then a vector that contains the count for each word in the vocabulary is used as the feature vector X for the regression model. We then train a logistic regression model for the prediction task. The logit function is $\beta_0 + \beta_1 X$, where β_0 and β_1 are model parameters. The reason why we choose logistic regression for this task is that we want a simple model that is good for a binary classification task as our baseline. We use `sklearn` implementation (Pedregosa et al., 2011) (v 0.22.2) for the model. The model outputs log probabilities of the book containing only German or German plus foreign languages.

For the second task, our goal is to predict which language the next character (for the DTA corpus) or the next word (for the LinCE corpora) would be in given a sequence of characters/words that have been read so far. Hence this is a seq2seq (Sutskever et al., 2014) model. Of all the choices we have for such a model, we use a character/word LSTM model as our baseline just for illustration purpose. The reason why we choose a character level model for the DTA corpus is that we want to simulate the environment for OCR where each character is sequentially scanned and predictions for the next character

Precision	Recall	F1
0.83	0.80	0.81

Table 3: Baseline results for the first task: book-level prediction.

Corpus/Input size	Precision	Recall	F1
DTA: 20 chars input	0.04	0.95	0.07
DTA: 50 chars input	0.13	0.81	0.22
DTA: 100 chars input	0.21	0.73	0.33
DTA: whole page input	0.63	0.13	0.22
LinCE: HIN-ENG	0.75	0.95	0.75
LinCE: SPA-ENG	0.79	0.81	0.79
LinCE: NEP-ENG	0.68	0.65	0.66
LinCE: MSA-EA	0.62	0.65	0.63

Table 4: Baseline results for the second task: sequential prediction.

are made as we proceed. We feed a text chunk (for the DTA corpus input, the chunk could contain 20, 50, 100 characters or characters from an entire page) to a character/word embedding layer (the embedding layer's output dimension is 16). We choose different chunk sizes so as to see whether the choice would influence overall task performance, and the conclusion could be helpful determining the optimal input size for OCR. For predictions of the DTA corpus, we concatenate the embedding layer output with the log probabilities output from the book-level predictions of the first task. We then send the resulting vectors to a single-layer LSTM of 16 hidden units. Then we take all hidden states output by the LSTM layer and send them to a linear layer with output dimension of 2. Then we apply a softmax function to the output vectors to obtain probability distributions over the two output classes. We train this model for 5 epochs. We report precision/recall/F1 for the DTA corpus with different input chunk sizes, and for the LinCE corpora with different language pairs.

In Table 3 and Table 4 we provide baseline results for the above two tasks. As we can see, the baseline model gives decent performance on the first task. For the second task, when having the DTA corpus as input, we see that as we increase the chunk size there is an increase in precision, but a decrease in recall. The F1 score first increases, then decreases as the chunk size increases. When we have the LinCE corpora as input, there is a general improvement in precision and recall. Furthermore, we see that the F1 scores for the DTA corpus input are significantly lower than for the LinCE corpus, which is probably due to the fact that the language distribution in the DTA corpus is much more skewed towards the matrix language than in the LinCE corpora, making the prediction inherently more difficult, like finding a needle in a haystack. This encourages improvement over the baseline model for future work.

5 Conclusion

In this paper, we study the code-switching patterns in 1,406 historical German books. We automatically annotate and manually inspect code-switched text spans. We then quantitatively show that the code-switching patterns in these books are different from those in typically studied informal code-switching domains, such as social media. We propose two interesting tasks that, if well handled, would help improve OCR performance on code-switched historical books. Finally, we provide baseline results for the two tasks. The first task gives decent baseline performance, and we see an obvious precision-recall tradeoff by input chunk size on the second task where we have the DTA corpus as our input. We also see that locating code-switching points is a more difficult task in historical text than in informal domains such as social media, since it occurs rarely in old books. The F1 score for the sequential prediction task could be further improved by utilizing different model architectures. We leave this for future work. Additionally, our study opens an avenue for more analysis of code-switching patterns in historical texts.

References

Gustavo Aguilar, Fahad AlGhamdi, Victor Soto, Mona Diab, Julia Hirschberg, and Thamar Solorio. 2018. Named entity recognition on code-switched data: Overview of the CALCS 2018 shared task. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 138–147.

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A centralized benchmark for linguistic code-switching evaluation. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 1803–1813.

Ruthanna Barnett, Eva Codó, Eva Eppler, Montserrat Forcadell, Penelope Gardner-Chloros, Roeland van Hout, Melissa Moyer, Maria Carme Torras, Maria Teresa Turell, Mark Sebba, Marianne Starren, and Sietse Wensing. 2000. The LIDES coding manual a document for preparing and analyzing language interaction data version 1.1—july, 1999. *International Journal of Bilingualism*, 4(2):131–132.

Jan-Peter Blom and John J. Gumperz. 1972. Social meaning in linguistic structures: Code switching in northern Norway. *In: John Gumperz and Del Hymes (eds): Directions in Sociolinguistics: The Ethnography of Communication*, pages 407–434.

Barbara Bullock, Wally Guzmán, and Almeida Jacqueline Toribio. 2019. The limits of Spanglish? In *Proceedings of the 3rd Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 115–121, Minneapolis, USA, June. Association for Computational Linguistics.

J.Y.C. Chan, P.C. Ching, Tan Lee, and Helen M. Meng. 2014. Detection of language boundary in code-switching utterances by bi-phone probabilities. In *International Symposium on Chinese Spoken Language Processing, IEEE*.

Khyathi Chandu, Ekaterina Loginova, Vishal Gupta, Josef van Genabith, Günter Neumann, Manoj Chinnakotla, Eric Nyberg, and Alan W. Black. 2018. Code-mixed question answering challenge: Crowd-sourcing data and techniques. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code- Switching*, pages 29–38.

Kwang-Il Goh and Albert-Laszlo Barabasi. 2008. Burstiness and memory in complex systems. *EPL (Europhysics Letters)*, 81(4):48002.

Gualberto Guzmán, Joseph Ricard, Jacqueline Serigos, Barbara E Bullock, and Almeida Jacqueline Toribio. 2017. Metrics for modeling code-switching across corpora. In *Proceedings of Interspeech 2017*, pages 67–71.

Chad Nilep. 2006. "Code switching" in sociocultural linguistics. *Colorado Research In Linguistics*, 19.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Sarah Schulz and Mareike Keller. 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*, pages 43–51, Berlin, Germany, August. Association for Computational Linguistics.

Royal Sequiera, Monojit Choudhury, Parth Gupta, Paolo Rosso, Shubham Kumar, Somnath Banerjee, Sudip Kumar Naskar, Sivaji Bandyopadhyay, Gokul Chittaranjan, Amitava Das, and Kunal Chakma. 2015. Overview of FIRE-2015 shared task on mixed script information retrieval. In *Working Notes of FIRE*, pages 21–27.

Vinay Singh, Deepanshu Vijay, Syed S. Akhtar, and Manish Shrivastava. 2018. Named entity recognition for Hindi-English code-mixed social media text. In *Proceedings of the Seventh Named Entities Workshop*, pages 27–35.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W Black. 2019. A survey of code-switched speech and language processing. *arXiv preprint*, arXiv:1904.00784.

Thamar Solorio and Yang Liu. 2008. Part-of-speech tagging for English-Spanish code-switched text. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1051–1060.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Jullila Hirschberg, Alison Chang, and Pascale Fung. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72.

R. Sprugnoli, Sara Tonelli, G. Moretti, and S. Menini. 2017. A little bit of bella pianura: Detecting code-mixing in historical english travel writing. In *CLiC-it*.

Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

David Vilares, Miguel A.Alonso, and Carlos Gómez-Rodríguez. 2015. Sentiment analysis on monolingual, multilingual and code-switching twitter corpora. In *Proceedings of the 6th Workshop on Computational Approaches to Sub- jectivity, Sentiment and Social Media Analysis*, pages 2–8.

Yoharshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. 2014. POS tagging of English-Hindi code-mixed social media content. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 974–979.

Jochen Weiner, Ngoc Thang Vu, Dominic Telaar, Florian Metze, Tanja Schultz, Dau-Cheng Lyu, Eng-Siong Chng, and Haizhou Li. 2012. Integration of language identification into a recognition system for spoken conversations containing code-switches. In *Spoken Lan- guage Technologies for Under-Resourced Languages*.