Universität
Zürich UZH

Master’s thesis
presented to the Faculty of Arts and Social Sciences
of the University of Zurich
for the degree of
Master of Arts in Linguistics
with concentration in Digital Linguistics

Code-switching detection in the Corpus of Early English Medical Writing (1375-1800)

Author: Eirini Valkana
Student ID Nr.: 20-728-135

Supervisor: Prof. Dr. Martin Volk

Department of Computational Linguistics
Submission date: 01.12.2022

Abstract

The aim of this thesis is the detection of Latin code-switches within the Corpus of Early English Medical Writing (1375-1800). Code-switching detection is a task of high importance in Computational Linguistics since many of the most common Natural Language Processing tasks, such as part-of-speech tagging and machine translation, strongly depend on the respective language. So far, there have been studies that focus mainly on code-switching detection in social-media data. However, only a few address code-switching detection in historical texts. The research questions posed in this thesis concentrate on the investigation of appropriate methods for the detection of code-switches within a historical corpus, and on the nature of the code-switches. After assessing the performance of two language models and that of a lexicon-based approach, the latter is chosen as the most suitable method for code-switching detection in the Corpus of Early English Medical Writing, yielding approximately 5,000 code-switching instances.[1]

[1] The code developed for this thesis can be found here: `https://github.com/EiriniVal/thesis/tree/main`

Acknowledgments

At this point, I would like to express my gratitude to the people to whom this thesis is dedicated. First of all, I would like to thank my supervisor Prof. Dr. Martin Volk who supported my idea from the start and guided me with patience and kindness throughout this exciting journey. Martin, I hope you enjoy reading the thesis! Additionally, I would like to thank my parents, Paraskevi and Kostas, and my brother, Dimitris, from the bottom of my heart for supporting me and believing in me my whole life through the ups and downs. Without them I would not be the person that I am today. I love you very much! I would also like to express my gratitude and love to my friends Sabrina, Vasiliki, Venetia, Fotis, Konstantina, Paraskevi and Dafni for being my craziest supporters and fans. Thank you for being the light in my life during the darkest times. I will always be there for you! Finally, I would like to thank my beloved Vasilis for his support and for believing in me more than I ever believed in myself. I love you! If you think this is over, it is not. I would ultimately like to thank my dog Bobby for being the cutest little puppy and for showing me how life without stress is like. Bobby, if you are reading this: I love you!
-Eirini

Σε αυτό το σημείο θα ήθελα να εκφράσω την ευγνωμοσύνη μου στα άτομα στα οποία αφιερώνω αυτή τη διπλωματική εργασία. Αρχικά, θα ήθελα να ευχαριστήσω τον καθηγητή και επιβλέποντα μου, κύριο Martin Volk, που υποστήριξε την ιδέα και την προσπάθειά μου από την αρχή και με καθοδήγησε με υπομονή και καλοσύνη κατά τη διάρκεια αυτής της συναρπαστικής διαδρομής. Martin, ελπίζω να απολαύσεις την διπλωματική! Επιπλέον, θα ήθελα να ευχαριστήσω από τα βάθη της καρδιάς μου τους γονείς μου, Παρασκευή και Κώστα, και τον αδερφό μου, Δημήτρη, που όλη μου την ζωή με υποστηρίζουν και πιστεύουν σε εμένα, στα εύκολα και στα δύσκολα. Χωρίς αυτούς δεν θα ήμουν ο άνθρωπος που είμαι σήμερα. Σας αγαπώ πολύ! Θα ήθελα επίσης να εκφράσω την ευγνωμοσύνη και την αγάπη μου στους φίλους μου, Σαμπρίνα, Βασιλική, Βενετία, Φώτη, Κωνσταντίνα, Παρασκευή και Δάφνη, που είναι οι πιο τρελοί και φανατικοί υποστηρικτές μου. Σας ευχαριστώ που φωτίζετε τη ζωή μου στις πιο σκοτεινές στιγμές. Θα είμαι πάντα εδώ για εσάς! Τέλος, θα ήθελα να ευχαριστήσω τον αγαπημένο μου Βασίλη για την υποστήριξη του και που πίστεψε σε εμένα περισσότερο απ΄ότι πίστεψα εγώ στον εαυτό μου. Σε αγαπώ! Εάν νομίζετε ότι τελείωσα, κάνετε λάθος. Καταληκτικά, θα ήθελα να ευχαριστήσω το σκυλάκι μου, τον Μπόμπι, που είναι το πιο γλυκό κουτάβι και που μου δείχνει πώς είναι η ζωή χωρίς στρες. Μπόμπι, αν το διαβάζεις: Σε αγαπώ!
-Ειρήνη

Contents

1 Off the beaten path; code-switching detection in the Corpus of Early English Medical Writing (1375-1800)

1.1 Conception of the idea

When I was called to decide on a thesis topic, I firstly looked what type of linguistic data I am interested in processing. My preference was to work with historical languages like Latin or Ancient Greek in order to investigate the progress that has been made in Natural Language Processing (NLP) with respect to historical languages. Most of the historical languages lack in corpora and NLP tools. However, according to Piotrowski [2012], the interest of the NLP community in historical texts and genres, other than the news wire texts, has started to grow. Apart from my interest in historical languages, I was also particularly interested in the linguistic phenomenon of code-switching, which is the alteration of language within a communicative event (a more detailed definition can be found in Chapter 1.3). Motivated by both the fact that there are not many NLP studies on historical languages, and the fact that the task of automatic code-switching detection is mostly limited to modern languages, I decided to combine both concepts and seek for methods to automatically detect Latin code-switches in texts, while also investigating the challenges that may arise from this task. The next step, in order to realise my thesis, was to search for data, and ideally a corpus, that would constitute a good candidate for this task. This implies that the candidate corpus should have a promising amount of Latin code-switches. After some research on the available corpora I came across the Corpus of Early English Medical Writing (1375-1800). The Corpus of Early English Medical Writing (CEEM) was compiled by Irma Taavitsainen and Päivi Pahta [Taavitsainen and Pahta, 2013], and it consists of medical texts written in English vernacular from 1375 to 1800. My assumption was that scientific texts from that era would possibly be good candidates for the task at hand, because Latin used to be the main language of science for many centuries.

This assumption was strengthened even further after I came across with related research on code-switching in Early English. More specifically, Schendl and Wright [2011] present a variety of instances of mixed-language texts from medieval and early modern Britain, for instance mixed poems and sermons which were rather negatively characterized by the medievalists as “macaronic”. Most importantly, it is clearly stated in [Pahta, 2012, p. 4] and [Schendl and Wright, 2011, p. 22] that code-switching was also very common in scientific texts, especially in medical texts. In fact, there are linguistic studies discussing code-switching in Old English and Early English data on the basis of mixing English with Latin and even with

French. However, quoting from [Schendl and Wright, 2011, p. 15], “it has taken a long time for the numerous mixed-language texts from earlier periods of English to be accepted as evidence for historical code-switching and thus worthy of serious linguistic investigation”. They also propose that “code-switching research should be extended into the historical dimension, hereby establishing historical code-switching as a new sub-field of historical linguistics”. All of the aforementioned statements reassured me that it would be possible to detect a significant number of Latin code-switches within the CEEM texts, and that this study will highlight the recently risen sub-field of historical code-switching thanks to the contribution of NLP.

1.2 Related studies on automatic code-switching detection

Among the NLP studies focusing on code-switching detection, only a few focus on historical data. The study of Volk et al. [2022] is one of them, and constitutes the main guideline for this thesis' main task of automatic code-switching detection. Volk et al. [2022] use a lexicon-based approach to perform automatic code-switching detection in the Bullinger corpus, which contains correspondence from the 16th century belonging to the Swiss reformer Heinrich Bullinger. The dominating mix of languages in the Bullinger corpus contains Early New High German and Latin. The Bullinger corpus is similar to CEEM in the sense that both corpora contain low-resource historical languages, namely Early New High German, Middle and Early Modern English, and Medieval Latin. This is of high importance because both corpora are expected to have similar challenges in processing. Therefore, the methods used on the Bullinger corpus can also be used on the CEEM. Hence, in this thesis I will be using an adapted version of this lexicon-based approach by Volk et al. [2022] to perform the task of automatic code-switching detection in the CEEM data. This approach will be described in detail in Chapter 3.2.

Schulz and Keller [2016] work with Latin-Middle English mixed texts as well, and hence, their study provided me with information on useful tools and methods for dealing with Middle English and Latin. In their work, they annotate macaronic sermons with language and part-of-speech information on the word level, with the goal of analysing code-switching rules within nominal phrases. Thus, they work with intra-sentential code-switching, which is code-switching within a span of a sentence (a detailed definition can be found in Chapter 1.3). In this paper, the authors use shallow features like Part-of-Speech, because there is a limited number of automatic processing tools for these languages.

Liu and Smith [2020] perform code-switching detection in historical German books in order to investigate the difference between code-switching in formal and informal contexts, such as social media. They present metrics for examining whether they can predict which books are possible hosts for code-switching, to improve the performance of the OCR task. As for the automatic code-switching detection they use information on the font of the text, and the off-the-shelf Detect Language API. Their study is very informative on the importance of code-switching detection task and on how challenging detecting code-switches can be when working with historical data, compared to modern language data.

Regarding code-switching detection in non-historical data, the studies are much greater in number, especially during the last decade. Most studies are related to modern language and social media data. The topics vary mainly between automatic code-switching detection, code-switching prediction and code-switching modeling.

Volk and Clematide [2014] describe the task of automatic code-switching detection in a corpus of Swiss Alpine texts, as well as the challenges that come with the task of language identification on the level of the sentence and especially on the level of the word. They test off-the-shelf language identification models, including LangID, which is also used in my study. They also exploit additional information such as text in quotation marks, and the parallel parts of the corpus written in other languages.

Another useful study for my thesis' background is the one by Martínez [2020]. In his thesis, he describes the process of obtaining corpora for code-switching. He is also elaborating on the tasks of part-of-speech tagging and language modeling, and addresses the issue of using monolingual resources for code-switching data.

The rise of Machine Learning has also raised the question on what kind of model architectures can be advantageous for performing such task. Various approaches have been proposed for code-switching detection. Zhang et al. [2018] use “CMX”, a feed-forward network with a decoder architecture, to label code-mixed texts. Jaech et al. [2016] focus on the word-level language identification and apply “C2V2L”, a hierarchical character-word model, to make language predictions on code-switched tweets. Samih et al. [2016] experiment with LSTM architecture and combine word and character embeddings to identify code-switching on Modern Standard Arabic-Egyptian and Spanish-English mixed texts. In Chapter 3.3, I present in detail different Machine Learning model architectures and features that are used for the task of code-switching detection. Then, I assess the advantages and disadvantages of those models specifically for the context of code-switching detection in the CEEM texts and, in general, on low resource languages that usually lack in corpora and resources.

To conclude, all the studies cited above strongly inspired my thesis' workflow by familiarising me with all the concepts, guidelines, and methods related to the task of code-switching detection.

1.3 Defining code-switching and explaining the importance of code-switching detection in the fields of Linguistics and Natural Language Processing

Code-switching, also referred to in the literature as code-mixing, language mixing, or bilingual speech, can be described as the change from one language or linguistic variety to another within a communicative event or within spoken or written discourse [Pahta, 2012; Schendl and Wright, 2011]. A code-switch may be a single word in another language, or a larger segment withing the discourse. It is a very common linguistic phenomenon that mostly appears in bilingual or multilingual communities [Schendl and Wright, 2011, p. 23], and it is very common particularly in spoken language [Schulz and Keller, 2016]. Pahta [2012] comments that the term is "variously defined" in the literature: Sometimes, code-switching can denote the change of language only at the sentence boundary and it is distinguished from code-mixing, which denotes the language alteration within sentence boundaries. However, in Pahta [2012], the term is used with the broader sense and refers to "any identifiable use of more than one language in the course of a communicative event". In this thesis, I will be using the term "code-switching" with this broader meaning.

Code-switching can be classified into two categories, inter-sentential and intra-sentential, according to its morpho-syntactic nature and the position it appears. On the one hand, inter-sentential code-switches are the ones appearing between sentences or independent clauses. On the other hand, the code-switches that appear within a sentence are called intra-sentential. Pahta [2012] also uses the term "extra-sentential code-switches" referring to the "freely distributable categories, such as interjections or discourse markers".

In my study, I exclusively refer to inter-sentential and intra-sentential code-switching. More specifically, if an entire sentence is recognized as Latin, I consider it as an instance of inter-sentential code-switch, given that all my texts' main language is English. Similarly, if there is one or more tokens that are recognized as

Latin within an English sentence, I consider them as instances of intra-sentential code-switching. Examples of the two categories can be found in Table 1. All the examples in the table are manually detected from the CEEM texts.

Sentences in CEEM	**Code-switching category**	**Source**
Explicit Prologus in Librum Vricrisiarum.	inter-sentential	Daniel, Liber Uricrisiarum 2. In Middle English Medical Texts (1375-1500)
þat ben whasschyn with þe jows ***Apium ranarum***.	intra-sentential	Agnus Castus. In Middle English Medical Texts (1375-1500)
Exempli gratia Every Brook will furnish him with Pebbles rugged brown and large enough which after his Brethren the other Physicians had condemned his Patient to un dergo the operation of Lithotomy he made him discharge by the sole virtue of Diureticks	intra-sentential	Armstrong, (1735). An Essay For Abridging The Study Of Physick. In Late Modern English Medical Texts (1700-1800)

Table 1: Examples of code-switching categories.

At this point, it is important to note that my approach on what I consider a Latin code-switch in the context of this thesis can be characterized as greedy. Many Latin words were inserted in the English language as code-switches. However, after frequent use they are established as loan words and as a part of the English vocabulary. If we deeply investigate the last example of Table 1, the phrase "exempli gratia" is originally Latin, but throughout the years it was established as a loan word. Nowadays, such phrases are regarded as loan words. However, in my approach such words will be treated as code-switches. Another particular case of code-switching, which is also present in the CEEM texts, is that Latin words appearing as code-switches within the English texts are actually latinized versions of Ancient and Medieval Greek words. For instance, in the Trilingual Herb Glossary included in the first sub-corpus of CEEM, I encountered words like "arnoglossa" or "arsenicon", which are Greek terms written in Latin alphabet.

Therefore, a stricter and more fine-grained approach would treat such words as particular cases of Greek code-switches. However, in the context of this thesis' greedy approach, such words are considered as Latin. I choose to follow this approach because I believe that the task of identifying Latin words within Middle English texts is challenging and requires improvement itself. However, once the task of code-switching detection yields satisfactory results, the foundations for a fruitful study on distinguishing between loan words and true code-switches will be laid.

1.4 The contribution of code-switching detection to Linguistics

Besides my strong interest in historical languages and code-switching, I am also convinced that such research is beneficial for numerous sub-fields of Linguistics. As regards historical linguistics, this study constitutes an important step towards the creation of digital corpora for historical code-switched writing, since there are not such corpora available yet. As Schulz and Keller [2016] note, "historical mixed text is an interesting, yet still widely unexplored, source of information concerning language use in multilingual societies of Medieval Europe". Additionally, detecting code-switching material from older periods can benefit the studies which examine the phenomenon diachronically from a structural and sociolinguistic point of view. [Schendl and Wright, 2011, p. 34]. As for the field of Computational Linguistics, Volk and Clematide [2014] highlight that the importance of the code-switching detection task lies in the fact that crucial Natural Language Processing tasks such as part-of-speech tagging, named entity recognition or parsing are language dependent. Schulz and Keller [2016] also explain that addressing the phenomenon of code-switching in Natural Language Processing is of great importance, because of the growing interest on developing methods close to real-world linguistic data. Therefore, developing tools to automatically detect code-switching is a prerequisite for successfully processing real-world linguistic data.

1.5 Research hypotheses and expected outcomes of the code-switching detection in the Corpus of Early English Medical Writing

The research aim and ultimate goal of this thesis is to automatically detect Latin code-switches, both inter-sentential and intra-sentential, within the texts of CEEM. To be more specific, I will first search the texts for inter-sentential code-switches, in other words, for sentences that are written in Latin exclusively. This means that the texts will be segmented into sentences, and based on this segmentation I will perform the inter-sentential code-switching detection. The last row of Table 2 contains a segment of an English text with the sentence in bold written entirely in Latin. In this example the code-switching happens on the broader sentence level. Therefore, this instance will be recognized as inter-sentential. The centre of interest will then move from the sentence level to the token level, as I will be searching for Latin code-switching entities within the corpus' sentences. The first example in Table 2 is an English sentence with a Latin code-switch. Note that even though there are two tokens recognized as Latin, namely "Salus" and "populi" they are counted as one intra-sentential code-switching entity. The second example in the table is a very long English sentence that consists of two code-switching entities of more than two tokens each. I should, therefore, consider in my approach that there are sentences with more than one intra-sentential code-switches. After performing the detection task, and as a secondary research goal, I will also investigate the performance of the code-switching detection task per code-switching category, namely inter-sentential and intra-sentential. In addition, at the first stage of my thesis, while pre-processing the texts, I intend to profile the corpus and extract information regarding its heterogeneity and linguistic features.

My first hypothesis is related to this heterogeneity of the CEEM texts. I expect to encounter a high number of types, also known as unique words, that are especially evident in the earliest sub-corpus, MEMT. My hypothesis is initially founded on the expected high spelling variation due to the variety of authors, the diachronic character of CEEM texts, which cover a span of almost five centuries, as well as the fact that in the earlier years a standard orthography was not established. This hypothesis is assessed while I perform corpus profiling in Chapter 2.3.

This thesis also assesses which method is more suitable for performing automatic code-switching detection in the CEEM texts. My goal is to first compare how language models perform on the task of language identification, and then assess the performance of a Lexicon-based approach proposed by Volk et al. [2022]. My hypothesis for this research question is that language models are not suitable for code-switching detection, especially when applying token level language identification. I support this hypothesis based on literature Volk and Clematide [2014]; Volk et al. [2022]; Zhang et al. [2018], where it is frequently stated that off-the-shelf language models perform poorly on the token level. In order to confirm this hypothesis on the CEEM data, I intend to evaluate two language models on data of sequences of various lengths. One of them is an off-the-shelf language model called LangID, and the other one is an implementation of a language model created by Samuel Läubli and Lenz Furrer which I will be training on my corpus.

Sentences in CEEM	**Number of Code-Switches**	**Type of Code-Switches**
And þis enoyntment is called **Salus populi** þe making of which shal be schewed afterward.	1	intra-sentential
as Aristotle saith in his booke of generacion, and corrupcion, **Ignis qui est in vltimo continentis, non est in fine, ebullitionis**, and in an other place, he saieth, **Elementa omnia adinuicem contrarietatem habent**, and that is well saied, for heate is an extreme contrarie to colde, so is moistenesse to drinesse, and more paine in the one apostumacion, then in the other.	2	intra-sentential
This is daily confirmed in Patients, that, though they are weakned for the present by every sort of artificial Evacuation, soon after find sensible relief and great ease. || **Ex conferentibus & lædentibus sumuntur indicationes.**	1	inter-sentential

Table 2: Code-switching detection examples.

My assumption is that the model which is trained on my data will perform better, since the CEEM texts include a lot of instances of domain-specific Medieval Latin terms, which may not be covered by a pre-trained on Classical Latin model. This evaluation takes place in Chapter 3.1. As for the lexicon-based approach proposed by Volk et al. [2022] I expect to yield a satisfactory amount of code-switches, since the two corpora share some of the challenges of medieval writing, and both contain Medieval Latin code-switches. While applying this method, I intend to discuss any advantages and drawbacks rising from applying the lexicon-based approach to the CEEM data. The lexicon-based approach for the detection in CEEM texts can be found in Chapter 3.2. In addition to that, in Chapter 3.3 I also comment on some of the Machine Learning methods proposed for code-switching detection, and I will propose which architectures, and at which stage of research, can be used for the detection task in corpora like the CEEM.

The final topic that I aim to discuss in the context of this thesis' topic, is related to the nature of code-switches in the CEEM. My hypothesis is that the majority of code-switches in the corpus are intra-sentential code-switches with a span of one or two tokens. I support this hypothesis based on the scientific nature of the texts, and the social and linguistic reality surrounding them, which gives me reason to believe that medical and botanical terminology included in the texts will frequently appear in Latin, and typically, such terms usually consist of one or two words. This is a question that is going to be answered last, after I manually evaluate the code-switching results produced by the lexicon-based approach in Chapter 3.2. More information regarding the social and linguistic reality of the CEEM texts will be presented in Chapter 2.1.

1.6 Thesis' Workflow

In the following Chapter 2, I introduce in detail the contents of the Corpus of Early English Medical Writing texts, and the influence that the social and linguistic reality of Medieval England had on them. In Section 2.2, I proceed to explain how I pre-processed and structured the texts before proceeding to the code-switching detection task. Then, in Section 2.3, preliminary insights from corpus profiling are presented, offering an all-encompassing image regarding the corpus' heterogeneity. In Chapter 2.4, I describe a particular case of text included in the corpus, namely the Trilingual Herb Glossary, the methods that I used to pre-process it and to detect herbs in the CEEM texts, as well as its possible contribution to the task of code-switching detection. In Section 2.5, the challenges that can arise in the task of code-switching detection are addressed. Next, in Chapter 3 I describe the methods that I used to detect the code-switches in the CEEM texts and share my final results. More specifically, in Section 3.1 I evaluate the two language models on sequences of various lengths to assess whether such models can be used for the task of code-switching detection in the CEEM data. Afterwards, in Section 3.2, I describe in detail the adapted version of the lexicon-based approach by Volk et al. [2022] that I use for the code-switching detection task. In the final Section 3.3, a literature search is conducted concerning the use of Machine Learning methods for the task at hand. Ultimately, in Chapter 4, I revisit my research goals and hypotheses while discussing the findings of this thesis, and I provide some insight regarding the advantages, drawbacks, and research opportunities resulting from this study.

2 Diving into the Corpus of Early English Medical Writing (1375-1800)

The Corpus of Early English Medical Writing (CEEM) was compiled by Irma Taavitsainen and Päivi Pahta in the Research Unit for the Study of Variation, Contacts and Change in English (VARIENG) of the University of Helsinki. It is a corpus consisting of medical texts written in English vernacular and it offers rich research material for the diachronic study of English as the lingua franca of science [Taavitsainen and Pahta, 2013]. CEEM was designed to provide a computer-readable and representative sample of medical texts written in England and covers the time span of roughly four centuries, from 1375 to 1800. It is divided chronologically into three sub-corpora. The first one is Middle English Medical Texts (MEMT) and it contains a sample of 86 texts, which cover the period from 1375 to 1500. The second sub-corpus is called Early Modern English Medical Texts (EMEMT), and consists of 354 [2] texts of the period from 1500 to 1700. The third sub-corpus is the Late Modern English Medical Texts (LMEMT), consisting of 628 texts from 1700 to 1800. The chronological order of the three sub-corpora is demonstrated in Figure 1.

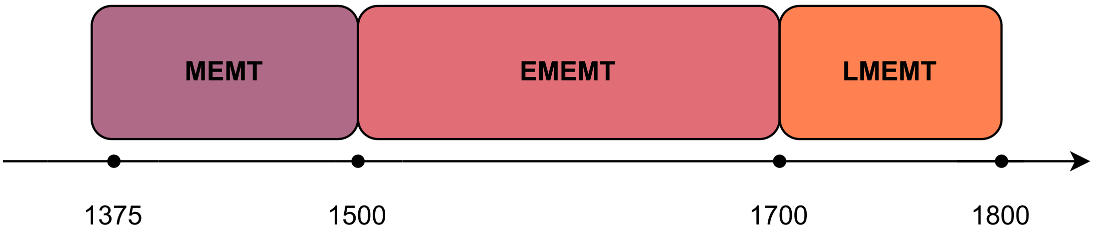


Figure 1: Chronological Order of the CEEM sub-corpora.

The texts of the corpus cover a wide variety of medicine-related genres, from informal remedy books to academic textbooks. CEEM contents include learned treatises, remedies, instructions for healthy living, theoretical treatises, and even surgical texts. Among the texts, there is a large number of medical recipes as well. Tables 3, 4, and 5 present the categories of the texts included within each sub-corpus. One should keep in mind that during the Middle Ages medicine was often associated with other domains like religion, botany, astrology, astronomy and philosophy. Therefore, many concepts from these fields intervene in this period's medical writing. One can often detect passages originating from ancient antiquity

[2] In their website they state that the number of texts is 450. However, this number is not only the raw texts, but it also includes the comments texts and some DUMMY.rtf files.

and mythology, or even segments from the Bible including prayers. Inserting such passages in the source language is also a mean for the authors to demonstrate their educational competence and their high social standing. Table 6 presents such examples from the CEEM texts.

MEMT
Specialized Texts
Surgical texts
Remedies and materia medica
Verse
Appendix: Trilingual Herb Glossary

Table 3: Contents of the MEMT sub-corpus.

EMEMT
Appendix: Medicine in society
Scientific journals
General treatises or textbooks
Texts on specific diseases, methods, substances, midwifery, children's diseases, and plague
Surgical and anatomical treatises
Recipe collections and materia medica
Regimens and health guides

Table 4: Contents of the EMEMT sub-corpus.

2.1 The reality surrounding the Corpus of Early English Medical Writing

As a starting point and as a foundation for the code-switching detection task, it is of great significance to first study the conditions around which the corpus' texts were developed, in order to grasp the linguistic particularities that may be encountered while processing the data of CEEM.

LMEMT
General treatises and textbooks
Specific treatises with four subcategories:
Diseases, Methods, Therapeutic substances, and Midwifery
Medical recipe collections
Regimens
Surgical and anatomical texts
Public health
Scientific periodicals:
The Philosophical Transactions and the Edinburgh Medical Journal
General periodical: The Gentleman’s Magazine

Table 5: Contents of the LMEMT sub-corpus.

2.1.1 The scientific reality

In England of late Middle Ages, a period when dramatic changes took place historically, scientifically and technologically, one can witness radical changes in language even in the relatively brief span of four centuries. Around 1500, the “medieval scholastic, logocentric science, relying on knowledge derived from Galen, Hippocrates and other ancient writers, gave way to new ways of constructing knowledge, relying on empirical methods and explanatory principles based on observation and cognition” [Pahta and Taavitsainen, 2011, p. 3]. The frustration of the scientific community with the outdated ancient medical methods led to scientific publications, which constitute landmarks of the medical writing and laid the foundation for the new innovative science of medicine. Pahta and Taavitsainen [2011] refer to such an important publication of the period in the field of anatomy written by Andreas Vesalius, with the title "De humani corporis fabrica" (1543). The scientific community started focusing more on the physiology of the human body, whereas until then the main focus was anatomy [Pahta and Taavitsainen, 2011].

Pahta and Taavitsainen [2011] mention that “thought-style” is a factor that influences scientific writing. Thought-style is defined as the “underlying scientific concepts, objects of enquiry, methods, evaluations and intellectual commitments related to the epistemology of science” [Pahta and Taavitsainen, 2011, p. 2]. This implies that changes in the scientific ideology, such as the ones aforementioned, reflect, and at the same time are reflected by, the linguistic features. Thus, a period

Segments from CEEM	**Domain**	**Source**
Venus with this herb healed her Son Æneas when wounded in the War. His words are Æneid. 12. Hic Venus indigno nati concussa a dolore; Dictamnum genitrix Cretæa carpit ab Ida, Puberibus caulem foliis & flore comantem Purpureo, non illa feris incognita Captis Gramina, cum tergo volucres hæsere sagittæ. About this time came in the Hungarian Infection, which was a Disease that bred such a putrefaction in the bodies of Men, that even when they were near death, they fell a vomiting but that with such a stench, that no body could endure it.	mythology	source: Minderer, (1686). Medicina Militaris. In Early Modern English Medical Texts (1500–1700).
For hors þat beo yfoundid. say þis charme .iij. and ask ate bygynnyng of whate colour þe hors is and þe name of hym þat it owe. And say þis in latyn: In nomine patris et filij et spiritus sancti. Amen.	religion	source: Medical Charms. In Middle English Medical Texts (1375–1500).

Table 6: Passages From Mythology and Religion Inserted in the Corpus in Source Language.

of great scientific changes entails alterations in the way such ideas are expressed.

In addition, other language-external factors, such as the level of the authors education, their social position, and the target audience, are also reflected in language use. Given that the authors of CEEM texts are numerous, I expected to encounter great linguistic variety in the CEEM texts.

Another key factor to consider is that the CEEM texts were written during a time when the discovery of printing technology changed the norm of knowledge transmission. According to Pahta and Taavitsainen [2011] medical publishing skyrocketed from 1640s onward. Thanks to the printing technology the authors of scientific texts could gain an income from their publications and be widely known. They started signing their texts with their own name, and their writing style became more polished to mirror their educational background and prestige.

2.1.2 The linguistic reality

Before proceeding to the linguistic reality of the CEEM texts in particular, I will briefly describe the eras of medical language, as described in Wulff [2004]. It is well known that Greek had been the main language of western medicine for centuries, and until today, this Greek legacy, as Wulff [2004] calls it, is evident in the terminology of modern medicine. Around the first century AD, Aulus Cornelius Celsus with his work "De Medicina", tried to give an overview of the medical knowledge that was originally written in Greek. Because most Greek terms did not have Latin equivalents, he tried to either import them directly, by also preserving their Greek grammatical endings, or he latinized them, by writing them with Latin letters and replacing Greek endings with Latin. He also tried to preserve the semantic richness of the Greek anatomical terminology by translating Greek terms into Latin, such as "dentes canini" from Greek κυνόδοντες , which means dog teeth. Thus, the Greek origins are sometimes evident even in the Latin terms. During the renaissance, Greek and Arabic works were translated into Latin, and according to Wulff [2004] this is when the Latin era of medicine language began, until the national languages prevailed.

The medical writing in medieval England reflects not only these different eras of the medical language in general, but it also mirrors the linguistic reality of medieval England in particular. As Schendl and Wright [2011] highlight, throughout history Britain has been a multilingual country. During the Middle Ages, Latin epitomised the language of religion, education, administration, law and literature. In most of these domains, Latin remained dominant even in the Early Modern period [Schendl and Wright, 2011, p. 18]. At the same time, different vernaculars were spoken throughout Britain. During the Old English period, the main vernacular was Old English, which was also in contact with Old Norse and Celtic languages [Schendl and Wright, 2011, p. 18]. After the Norman conquest of England in 1066, French and Latin gradually came to the fore and they were considered prestige languages [Pahta, 2012, p.3]. Latin used in Britain was influenced by Anglo-Norman French, and by the late fourteenth century, a mix of these three languages was used. According to Taavitsainen and Pahta in Taavitsainen and Pahta [2009], the first phase of the vernacularization of science and medicine started around the fourteenth century. The second and third phase commenced a century after, from 1500 to 1700 [Pahta and Taavitsainen, 2011, p. 3]. Gradually, English medical writing started replacing Latin. Finally, by the 1700s English became the governing language of medical writing in England [Pahta and Taavitsainen, 2011, p. 4]. Proof is that around the same period the Royal Society of London was founded and published the scientific periodical of Philosophical Transactions (PT) mainly in English [Pahta and Taavitsainen, 2011, p. 4]. Nevertheless, such shift was not steady and there were periods when the use of Latin was still preferred.

Regarding CEEM, the language witnessed in the first sub-corpus is mostly Middle English. Around 1500, the "periodization [3] of the English language from Middle English to Early Modern English" [Pahta and Taavitsainen, 2011, p. 20-22] starts. The shift and continuity of the English language can be traced in the second sub-corpus, EMEMT, in which Early Modern English is dominant. Subsequently, the gradual shift to Late Modern English is evident in the third sub-corpus, LMEMT. Due to this multilingual reality just described, the texts of CEEM consist of many individual words or entire passages in other languages, mainly in Medieval Latin, Greek, and Anglo-Norman. One should keep in mind that many of these medical texts are translations of texts written in other languages, mainly Latin. One of the most famous examples is Chauliac's “Chirurgia Magna”, written originally in Latin. The numerous English versions of Chauliac's work reveal the effort of the authors-translators to establish medical terminology in Middle English [Schendl and Wright, 2011, p. 8]. It is important to mention that medieval mixed texts such as the ones in CEEM do not consist evidence of the authors' incompetence. In contrast, they reflect the multilingual competence and identity of theirs and their audience [Schendl and Wright, 2011, p. 20]. The writers may be unwilling to translate certain passages and words for numerous reasons. For instance, certain words and phrases may not have an English equivalent, or they may come from a well-known source like the Bible and they constitute a common knowledge even in their form source language. Furthermore, the language of the corpus has all the typical orthographic features of medieval writing. To be more specific, it is characterized by the three indicative features of historical language, as they are mentioned in [Piotrowski, 2012, p. 12-13], namely spelling difference, spelling variance, and uncertainty. Spelling difference means that the spelling of the words differs compared to today's spelling. Spelling variance refers again to a variety in spellings that even appear during the same period, mainly because the standard orthographies had not been established yet. As Piotrowski [2012] explains, spelling difference is related to diachronic variation, whereas spelling variance is

[3] Division of the history into periods

related to synchronic variation. The last feature described by Piotrowski [2012], uncertainty, refers to the fact that such medieval texts may contain errors, mainly because they have been digitized automatically by Optical Character Recognition, and they have been transcribed numerous times. Table 7 illustrates examples on how these features are reflected in the CEEM texts.

Medieval Writing Features	**Standard Form**	**Form in CEEM**
Spelling Difference	that	þat
Spelling Variance	sick	seke, sek, syke, syk, sike, sik
Uncertainty	Καχεξία	Καηεξια

Table 7: Spelling features of medieval writing as reflected in the Corpus of Early English Medical Writing.

2.2 Harmonizing the formats and structure of the CEEM texts

The kick-off of any Natural Language Processing task after gathering relevant data is pre-processing. Usually, in NLP tasks the data are raw unstructured texts in diverse formats that may or may not facilitate automatic processing. The goal of pre-processing is to prepare the data by structuring the information that is useful and related to the study at hand, while ignoring impractical information. Therefore, I first have to elaborate on the type of data I want to keep, and those I can discard. Usually, the most common pre-processing techniques are lowercasing, punctuation removal, tokenization, lemmatization, Part-of-Speech tagging and normalization. However, depending on the task at hand the researcher should choose the techniques which are appropriate for their own task and data.

2.2.1 Deciding on the File Format

As regards the CEEM data and the task of automatic code-switching detection, I had to first decide on a common file format for all the texts. Extensible Markup Language (XML) seemed ideal for this task because the goal is to have rigorously structured data that can also be individually labeled and searched.

Due to the fact that the three sub-corpora of CEEM were not compiled simultaneously, there is a variety regarding the format and structure of the files. To be more specific, the MEMT sub-corpus comes with the text files and the corresponding meta-information files in Rich Text Format (RTF). RTF which was developed by Microsoft and it is used for documents on their products. The second sub-corpus, EMEMT, the text files come in Text Format (TXT), which is more easily processable compared to RTF, but it still leaves the text unstructured. For each text there is usually an additional HTML info file containing meta-information, a TXT file containing annotation comments, a TXT file containing a normalized version of the text, and a TXT file containing annotation comments on this normalized version. Finally, the texts in LMEMT are distributed into two different versions: the digital version, in which the texts come in XML format, and the unannotated version, in which the texts come in TXT format. In the digital version, the TEI XML schema is used. TEI guidelines allow the insertion and structure of text information, especially meta-information, and it is commonly used in large collections of corpora.

For consistency reasons and in order to process the data all at once, I first convert the RTF files to a simple TXT format, such that all texts in the corpus have at least one common file format. For this task, I use a python library called striprtf and with a single loop over the texts of MEMT I successfully convert all RTF files to TXT format.

At this point, all the files of all three sub-corpora have at least one format that is TXT. In order to proceed to the XML conversion, it is essential to first consider the structure of the XML files. My goal is to create an XML structure that is a similar -yet more simplified- version of the TEI schema. When using the TEI schema, it is a common practice to also include metadata information inside the $< teiHeader >$ node. Similarly, in the CEEM XML files that I generate the XML structure starts with the root node called $< text >$. The $< text >$ node has two immediate child nodes, namely the $< header >$ and the $< contents >$. The $< header >$ contains metadata information about the name of the author, the title, the year, the volume, and the pages of the text. If any of this information is not detected, the content of the respective node is represented by a dash

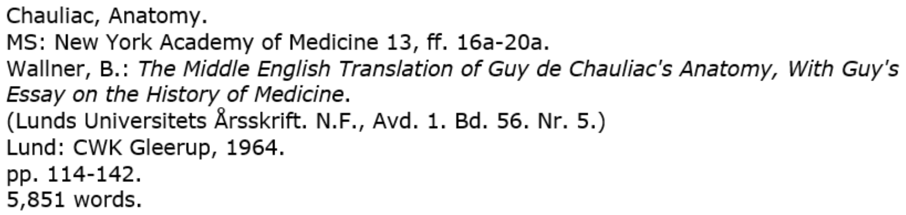
(-). Subsequently, the $< content >$ node covers the main text. As described in Chapter 1.5, due to the fact that the code-switching detection will take place both on the level of sentences (inter-sentential), and on the level of the words (intra-sentential), the data should be segmented into sentences, and subsequently, into tokens, also known as running words. Thus, the $< content >$ node, consists of multiple $< sentence >$ children nodes, which consist of $< token >$ children nodes. In order to keep track of the tokens and sentences per file I assign every sentence and token in the with a unique “id” attribute. Some of the texts include page information, which is also saved into a $< page >$ child node of the $< content >$ parent node. The $< page >$ node may appear inside a $< sentence >$ node, if the sentence is split between two pages in the original manuscript.

2.2.2 Gathering meta-information from the CEEM Texts

After having decided on the XML file structure, I deal with the task of extracting the metadata from the texts. Once again, due to the diversity of the conventions used on file-naming and on the structure of each sub-corpus, different techniques are used for accessing the relevant metadata from each sub-corpus.

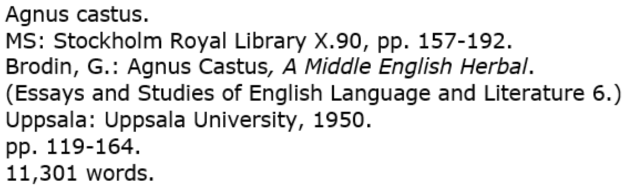
In the MEMT corpus, each text is accompanied by another TXT “info” file, which consists of meta information regarding the author, the title, the original manuscript and the pages. One disadvantage of the MEMT texts is that there is no year information for most of the texts (refer to Figure 3). It is only known from the introduction in the corpus that these texts were written around the 14th and 15th century. The authors of many of the MEMT texts are also unknown, with some exceptions like in Figure 2. To access the metadata of the MEMT texts, I use a python function which reads the first line of each information file to access the name of the author -if present- and the title of the text. Then, I extract the page information by locating the line that starts with "pp.". Figures 2 and 3 demonstrate the contents of two information files. The first one includes information regarding the author and the title in the first line, in the form of “Author, Title.”. However, in the second example there is no author information and only the text’s title is included in the first line. Both instances, contain page information and information regarding the original manuscripts.

Regarding EMEMT, I decided that it would be more practical to extract the meta information from the filenames, rather than the HTML documents. More specifically, each filename contains information on the year, the author, and the title of the text. Moreover, the filenames of the category “Philosophical Transactions” include additional information regarding the volume and the pages of the



Chauliac, Anatomy.
MS: New York Academy of Medicine 13, ff. 16a-20a.
Wallner, B.: *The Middle English Translation of Guy de Chauliac's Anatomy, With Guy's Essay on the History of Medicine*.
(Lunds Universitets Årsskrift. N.F., Avd. 1. Bd. 56. Nr. 5.)
Lund: CWK Gleerup, 1964.
pp. 114-142.
5,851 words.

Figure 2: Example of meta-information file in MEMT.



Agnus castus.
MS: Stockholm Royal Library X.90, pp. 157-192.
Brodin, G.: Agnus Castus*, A Middle English Herbal*.
(Essays and Studies of English Language and Literature 6.)
Uppsala: Uppsala University, 1950.
pp. 119-164.
11,301 words.

Figure 3: Example of meta-information file in MEMT without author.

text. For this reason, I gather all the metadata from the filenames using python, I specifically extract information on volume and pages if the text belongs to the category of Philosophical Transactions.

The third sub-corpus, LMEMT, has different file-naming conventions depending on the category of the texts. It would therefore be inefficient to extract meta-information from the filenames. Thus, I take advantage of the digital version of the texts and extract the metadata from the XML files rather than the filenames, such that I can process the categories all at once. The only meta information that I extract from the filenames is the year. I locate the XML nodes that contain the targeted information inside the $< teiHeader >$ parent node. More specifically, after locating the node $< sourceDesc >$, I access the subnodes $< persName >$, for the author's name, and $< title >$, for the text's title. Then I locate the node $< biblScope >$ and access the volumes if $unit = vol$ is present, and the pages if $unit = page$ is present.

2.2.3 Sentence segmentation and tokenization of the CEEM texts

After gathering the metadata, the goal is to structure the actual raw text into the XML node < *content* >. All three sub-corpora, contain special annotation symbols enclosing meta-information regarding the text's content, namely the title of the text or the chapters, the original manuscript, and also comments added by the transcribers for missing characters and omitted phrases. Table 8 describes the various uses of those symbols in the CEEM texts. The annotation symbols with left and right curly brackets are removed, but their content is preserved since it contains raw text. The rest of the annotation symbols depicted in Table 8, are removed together with their content, since the infomation of title and author is already included in the header of the XML files, and the rest of the information regarding the manuscript is not significant for the task at hand. At this point I also remove the blank lines from the texts. Then I concatenate all the lines of the file into one single string in order to pass it as a whole to the sentence segmentation and tokenization pipeline. Additionally, I changed the encodings of all files to UTF-8, given that some of them were in WINDOWS-1252, also known as CP-1252, in order to ensure consistency among the files.

Annotation Symbols	
[}...}]	chapter title
[{...{]	missing characters
[\...\]	manuscript information
[/.../]	paragraph
[^...^]	title, author, manuscript information, omitted words

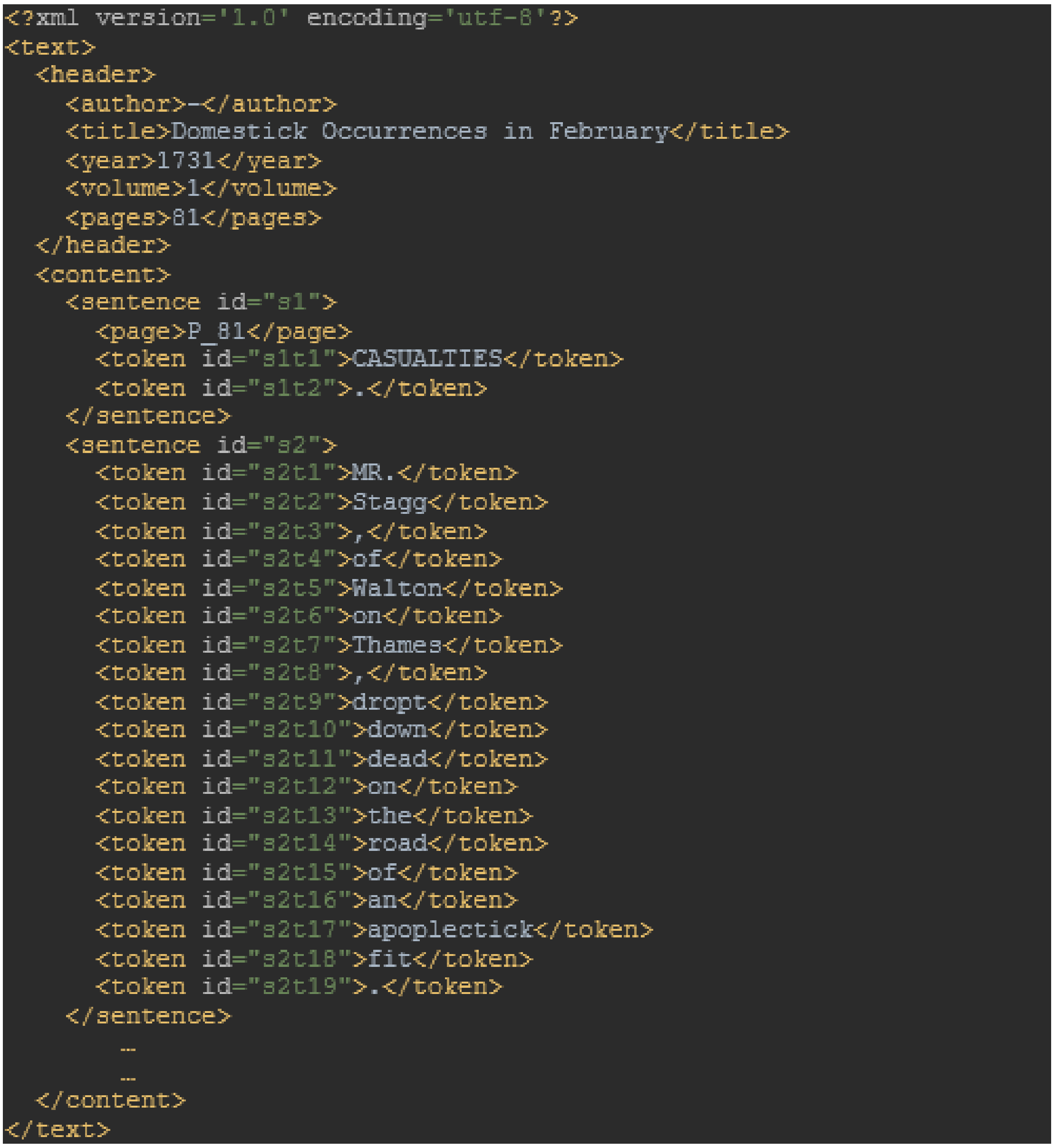
Table 8: Annotation symbols and their use in CEEM.

After extracting the raw text from the files in a single string, I had to find a suitable NLP tool to segment the string into sentences and tokens, and then structure it into the XML file, as previously described. One of the issues when dealing with languages like Middle English and Medieval Latin is that many of the well-established tools used by the linguistic community are trained on modern languages and perform poorly on low-resource languages. This is because of the different conventions in the orthography of medieval writing as described in the third Chapter of Piotrowski [2012]. As he describes, orthography does not only refer to the spelling of individual words, but it also touches upon issues like the use of punctuation, abbreviations, hyphenation, and the separation of words [Piotrowski, 2012, p. 11]. All of these features are usually crucial for both the tokenization and the sentence segmentation. For example, in some earlier texts of the CEEM, punctuation marks are missing or they are scarcely used. Furthermore, many texts of the CEEM contain scribal abbreviations, also known as sigla, that constitute single tokens, but they are erroneously split apart by NLP tools like spaCy. Thus, for the segmentation of the CEEM data into tokens and sentences I decided to use the Natural Language Toolkit (NLTK) package.

Ultimately, I arrange the segmented text into the XML nodes inside the $<$ *content* $>$ node of the XML file. For each sentence and token in the text I assign a unique index that will facilitate the search of code-switches within the large number of texts included in CEEM. Figure 4 displays a part of such an XML structured text.

2.2.4 Deciding on normalization using VARD

As mentioned in the beginning of this chapter, one common pre-processing task in NLP is normalization, especially when one deals with data that are characterized by high spelling variation, like the language of CEEM. The goal of normalization is to reduce this variation by generating a standard form of each word and subsequently, to improve the performance of other tools such as Part-of-Speech taggers. After investigating the topic and the tools that could be used for the CEEM texts, I came across the comparison study of Schneider [2020], who compare Statistical Machine Translation techniques with VARD for the normalization of Late Modern English. VARD is an interactive software designed for the normalization of historical texts, most commonly in Early Modern English, that are characterized by great spelling variation. The user can process the texts both manually, by setting candidate replacement pairs, and automatically, allowing the system to use the best candidate replacement found. There is also the possibility of training the tool on a sample of corpora. I tested the normalization tool on my own data. However, in many cases and especially in texts coming from the first sub-corpus, VARD did not perform well. In such cases, in order to improve the normalization results, it is recommended to set a specific threshold as the minimum confidence score, such that the generated normalized spellings have a higher confidence score. Given that there are numerous normalization suggestions for each word, if the set threshold is not reached by the top normalization suggestion, then the word is left as a variant. I tried different thresholds for my data but I realised that, in every setting, many words were substituted with the wrong normalized forms. Table 9 illustrates how VARD normalized specific words from the corpus. One interesting finding is that the Latin phrase “astrologia longa” was also normalized to English. Another issue was that the scribal abbreviations were not automatically normalized, and the only solution was to manually set the candidate replacements for



```
<?xml version='1.0' encoding='utf-8'?>
<text>
  <header>
    <author>-</author>
    <title>Domestick Occurrences in February</title>
    <year>1731</year>
    <volume>1</volume>
    <pages>81</pages>
  </header>
  <content>
    <sentence id="s1">
      <page>P_81</page>
      <token id="s1t1">CASUALTIES</token>
      <token id="s1t2">.</token>
    </sentence>
    <sentence id="s2">
      <token id="s2t1">MR.</token>
      <token id="s2t2">Stagg</token>
      <token id="s2t3">,</token>
      <token id="s2t4">of</token>
      <token id="s2t5">Walton</token>
      <token id="s2t6">on</token>
      <token id="s2t7">Thames</token>
      <token id="s2t8">,</token>
      <token id="s2t9">dropt</token>
      <token id="s2t10">down</token>
      <token id="s2t11">dead</token>
      <token id="s2t12">on</token>
      <token id="s2t13">the</token>
      <token id="s2t14">road</token>
      <token id="s2t15">of</token>
      <token id="s2t16">an</token>
      <token id="s2t17">apoplectick</token>
      <token id="s2t18">fit</token>
      <token id="s2t19">.</token>
    </sentence>
       …
       …
  </content>
</text>
```

Figure 4: Example of a text structured in XML format.

each one of them. Consequently, assuming that not normalizing would not affect the task of code-switching detection, and given that normalization of Middle English is a challenging task which requires special research itself, I ignored the task of normalization for the code-switching detection.

Original Spelling	VARD Normalized Spelling	True Normalized Spelling
erbis	orbs	herbs
wyl	will	✓
blod	blot	blood
preste	pressed	priest
astrologia longa	astrology lunge	no normalization

Table 9: Spelling normalization instances using VARD.

2.3 Preliminary insights from corpus profiling

2.3.1 Counting tokens and types

After segmenting and structuring the texts of CEEM, the next goal is to investigate the variation within the corpus by counting the tokens and the types of each sub-corpus individually and in total. The results of the counts are depicted in Table 10, and the token percentages over the whole corpus are depicted in Figure 5. In order to get the type counts, I use three different methods of varying strictness. The first method is to count all unique words appearing in the sub-corpora and the CEEM in total, including modern and roman numerals, while at the same time preserving the original casing of the words. These counts are included in the "types_all" column. The second stricter method, depicted in the fourth column of the table, excludes all the numerals, both modern and roman, but it preserves the original casing of the words. Finally, using the third and strictest method, I remove all the numerals and lowercase all the words to get the type counts. Given that the corpus is designed for examining the diachronic variation of English within a period of great linguistic changes as the ones described in the beginning of Chapter 2, I expected to encounter a high type count, that is particularly higher in the older texts of MEMT.

Regarding the size of the sub-corpora in tokens, it is clear from the pie chart that EMEMT and LMEMT have approximately the same size, which is about 45% of the total token counts. MEMT is the smallest in size, only representing 10.8% of the total token counts. Moreover, the counts in Table 10 confirm my hypothesis described in Chapter 1.5 regarding the heterogeneity and great spelling variation of the CEEM, and MEMT in particular. Looking at the fifth column of the table, it is evident that the types of MEMT constitute the 7.8% of the

corpus	tokens	types_all	types_w/o_numbers	types_strict
MEMT	551,600	46,019	45,933	43,404
EMEMT	2,275,592	97,450	97,035	81,944
LMEMT	2,267,941	71,489	70,428	56,364
Total	5,095,133	176,008	174,810	147,512

Table 10: Token and types counts of the CEEM.

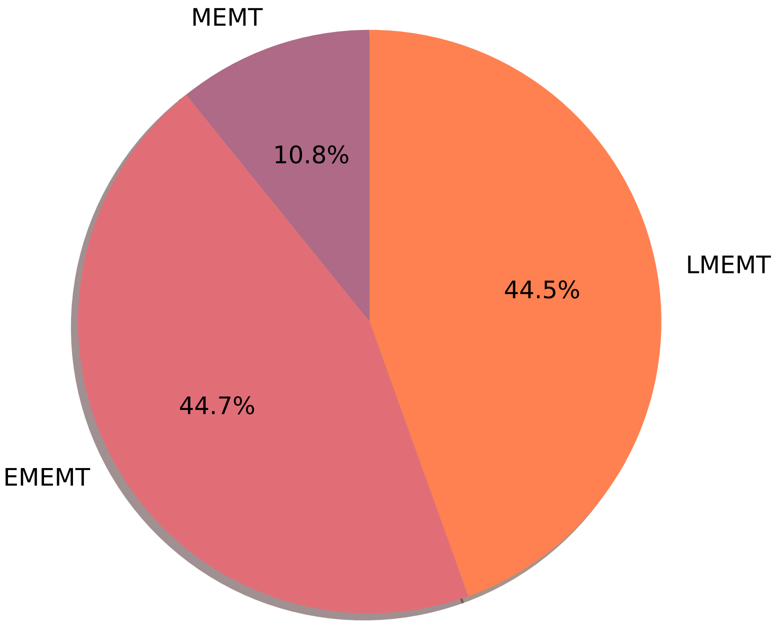


Figure 5: Token percentage over the whole corpus.

MEMT sub-corpus, whereas the types of EMEMT and LMEMT constitute 3.6% and 2.4% respectively. The bar chart depicted in Figure 6 demonstrates that there is a significantly higher spelling variation in the first sub-corpus compared to the following two, even if it has the smallest size in tokens. The language of MEMT is still not close to the standard Modern English and there a standard othography

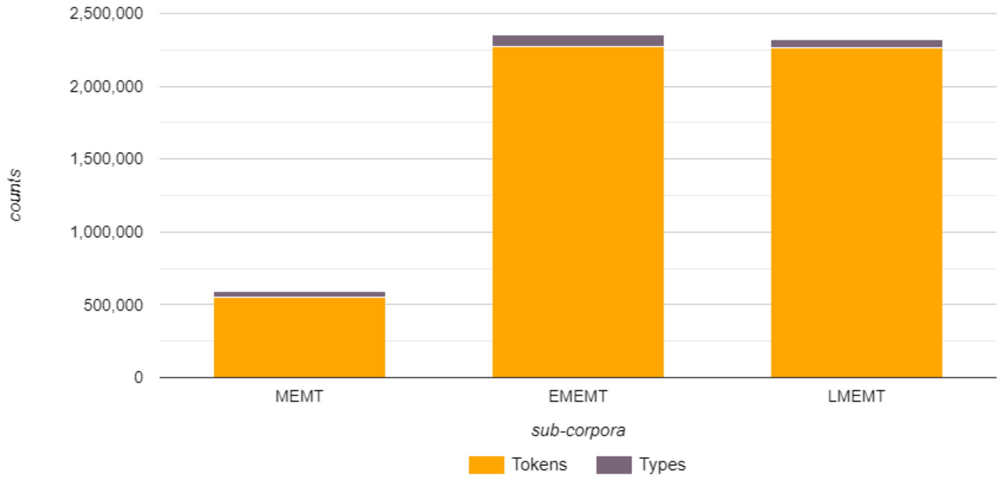


Figure 6: Comparison of the token and type counts within the three sub-corpora of CEEM.

is not yet used and established. Therefore, the percentage of types out of all the tokens per sub-corpus is larger for the first sub-corpus, and gets smaller for the next two sub-corpora. Additionally, the total number of tokens and types in the corpus shows that it can provide a solid foundation for a fruitful code-switching detection research.

2.3.2 Special characters

After examining the corpus size, it is time to investigate another interesting feature of the "Corpus of Early English Medical Writing", namely the special characters in the corpus. After a first glance at the texts I realized that there are a few occurrences of Old English alphabet letters. However, I soon realised that these were not the only special characters in the corpus.

Given that an important part of the corpus' contents is medical recipes, I discovered that certain texts that contain recipes include special symbols that were used by apothecaries to denote recipe related concepts. Table 11 shows all the special characters that I encountered.

Special Character	Name	Apothecary Symbol	Meaning
ȝ/3	yogh	ʒ	drachm/dram
æ	ligature of a + e/ash	℥	ounce
ð	eth	℞	recipe
þ	thorn		
ƿ	wynn		
œ	ligature of o + e		

Table 11: Special characters in the CEEM texts.

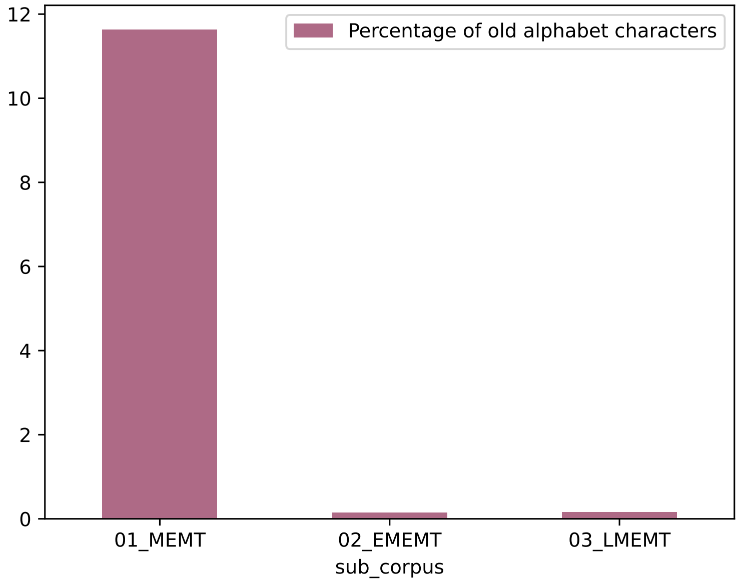


Figure 7: The percentage of special characters per sub-corpus.

At this point, I decided to compute how often words with such characters occur in the corpus, with the aim to assess their impact on my research. Therefore, when searching for tokens with special characters in the corpus in addition to the old alphabet characters, I also consider the apothecary symbols. For this task I use regular expressions on the token level. It has to be noted that number “3” is most commonly used within the tokens to represent the letter “yogh”. Therefore, I also use a regular expression to match this specific pattern. Of course with such an approach tokens like "3rd" are inevitably considered. Nevertheless, because ordinal numbers with numbers and letters combined do not appear that frequently in the texts, I assumed that it would be useful to actually include tokens including number "3". The percentage of the tokens with special characters is depicted in Figure 7. It is obvious from this chart that throughout the years the use of old alphabet characters is significantly decreased.

2.3.3 Scribal abbreviations

Another particularity of the language of CEEM, and of medieval writing in general, is the use of scribal abbreviations, also called sigla. These abbreviations were mainly used in ancient and medieval times for efficiency reasons when copying or writing manuscripts and because of the expense of parchment and paper [Reimer, 1998]. In this corpus most of the scribal abbreviations used in the original manuscripts are characters in superscripts, or a macron symbol over certain letters which denotes a missing "n" or "m". In the digitized version of CEEM, the compilers transcribed the letters that are originally in superscript by surrounding them with two equal "=" signs. The macron was transcribed as a tilde "∼" symbol next to the respective letter. Some frequently occurring abbreviations in the CEEM are demonstrated in Table 12.

Original	**CEEM**	**Meaning**
y^{t}	y=t=	that
w^{t}	w=t=	with
y^{e}	y=e=	the/you
drynk$\bar{y}$	drynky∼	drinking
$\bar{\imath}$	i∼	in

Table 12: Scribal abbreviations in CEEM.

Figure 8 depicts the percentage of scribal abbreviations out of all the tokens of each sub-corpus. The LMEMT sub-corpus scores the lowest. My explanation is that after the 1700s the use of sigla was completely abandoned because printing technology was already popular and at the same time the standard orthographies started to gain ground. The EMEMT sub-corpus scores the highest. My hypothesis is that this increase is due to the urgent need of writing and translating medical texts in a period when new medical approaches and ideas constantly emerged. Such abbreviations would help the scribes save time, especially during a period when printing medical texts had just started to gain a lot of popularity.

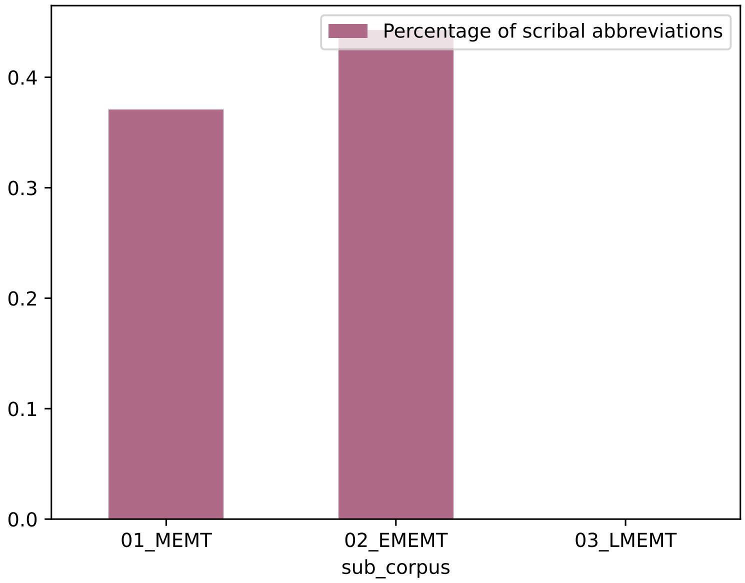


Figure 8: The percentage of scribal abbreviations per sub-corpus.

2.3.4 Numerals

The last stage of corpus profiling is to find the percentages of numerals in the corpus. The two types of numerals that I considered are both modern (arabic) and roman numerals. For this task I used pattern matching on the token level. As for the modern numerals, I also consider decimal numbers, ordinal numbers written with at least one number and the suffixes -st, -nd, -rd, -d, -th, and fractions with a slash “/” symbol. For the roman numerals, I also include in the research pattern the medieval variations of the Latin numerical system. The first one is the use of "j" instead of "i" when in the final position. For example, number three can be written as “iij” instead of “iii”. The second difference is that in the medieval variation of the Latin numerical system, the additive notation was used instead of the subtractive notation. For instance, the number four could be written as “iiij” instead of “iv”.

Roman numerals were still popular until the 14th century, but gradually they got replaced by the equivalent arabic numbers. This gradual substitution is evi-

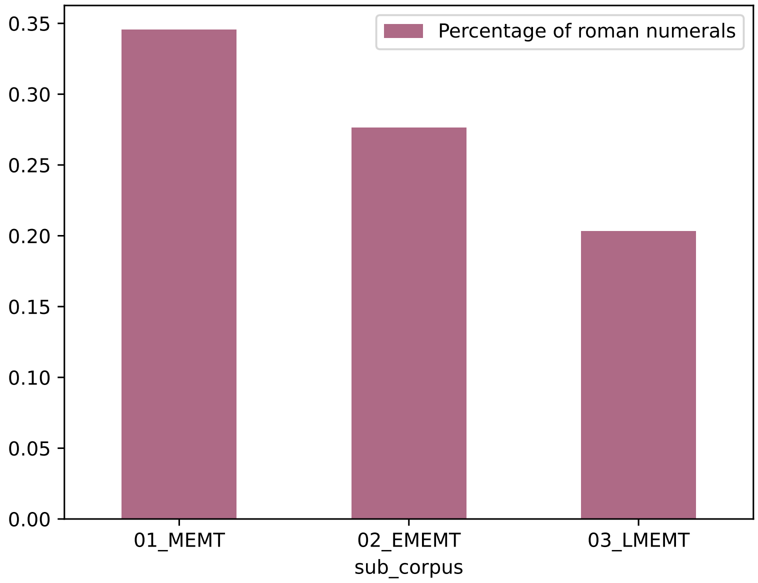


Figure 9: The percentage of roman numerals per sub-corpus.

dent when comparing the Tables 9 and 10.

2.4 Trilingual Herb Glossary

2.4.1 Exploring and structuring the glossary's contents

Among the texts of the first sub-corpus of CEEM I came across a different type of text, namely the Trilingual Herb Glossary. Because of its structure, I decided that it would be meaningless to process it as the rest of the texts, but it would rather be suitable to choose another file format for structuring its contents. Before explaining the processing of the glossary, I will briefly give an overview of its structure and content.

The Trilingual Herb Glossary is originally included in the manuscript BL Sloane 146, ff. 69v-72r, and it was digitised [4] by the British Library. This manuscript

[4] British Library Sloane MS 146, http://access.bl.uk/item/viewer/ark:/81055/vdc_100063649028.0x000001

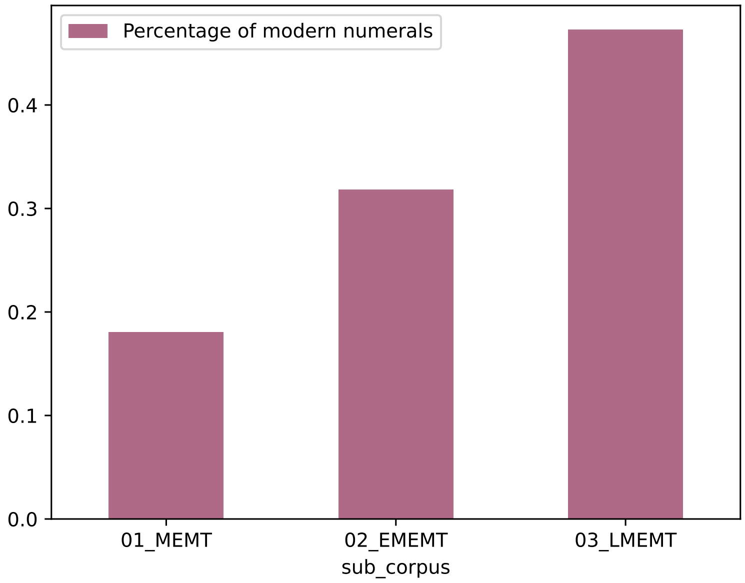


Figure 10: The percentage of modern numerals per sub-corpus.

was written in the 13th century, and in addition to medical and botanical texts, it includes the botanical glossary which contains words in three languages, namely Latin, Anglo-Norman, and Middle English. The glossary has been written in the late 12th century. Here the word “glossary” is used with the sense of vocabulary. More specifically, the glossary terms are written in Latin, and they are usually followed by their translations in Middle English and Anglo-Norman. Figures 11 and 12 illustrate the same section from the glossary. Figure 11 depicts the original manuscript and Figure 12 shows the same segment as it appears in the CEEM. Nevertheless, it is clear from Figure 11 that some entries are written in only one language. Another interesting feature of this glossary is that the first few entries are not alphabetically ordered, but after a certain point, the entries are structured in alphabetical order that only reaches to letter “c”. This peculiarity exists in the original version as well. The last feature of this glossary, that should be taken into account in the upcoming glossary-processing step, is that in some cases a herb name is represented by more than one names per language.

I decided to structure the data into a JSON format file, because it is easily readable and firm. In order to achieve this, I first have to retrieve the informa-

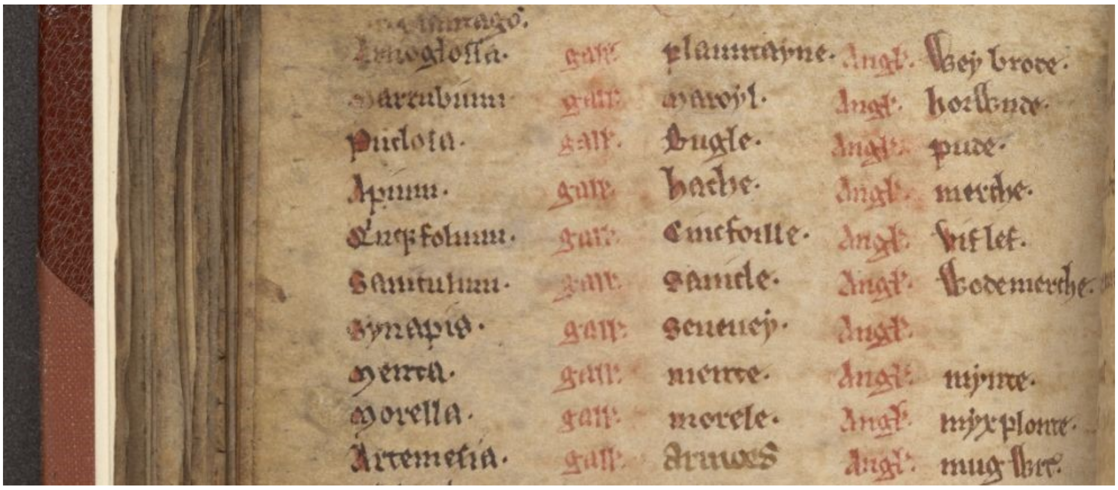
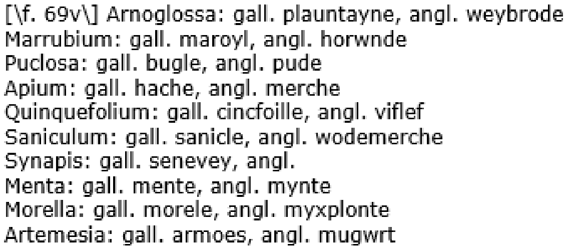


Figure 11: Fragment from the Trilingual Herb Glossary, as preserved in Sloane MS 146, ff.69v.



[\f. 69v\] Arnoglossa: gall. plauntayne, angl. weybrode
Marrubium: gall. maroyl, angl. horwnde
Puclosa: gall. bugle, angl. pude
Apium: gall. hache, angl. merche
Quinquefolium: gall. cincfoille, angl. viflef
Saniculum: gall. sanicle, angl. wodemerche
Synapis: gall. senevey, angl.
Menta: gall. mente, angl. mynte
Morella: gall. morele, angl. myxplonte
Artemesia: gall. armoes, angl. mugwrt

Figure 12: Fragment from the Trilingual Herb Glossary, in the CEEM.

tion contained in the glossary. This means finding all the different names for each language that correspond to a single entry. In order to do this automatically, I locate indicators like abbreviations denoting the language, e.g. angl., or punctuation marks like the colon, or the comma in order to efficiently structure the information provided for each entry. One of the challenges of this task is that the entries in the glossary are not written in a consistent way, and as a result, more than one indicators are used to denote the same thing. Table 13 illustrates some of these inconsistencies. For instance, the Middle English version is indicated by both "ang." and "angl.". The different names per language are sometimes separated by a slash while for other entries they are separated by the word "oþer" (meaning "or" in Middle English), or even the word "vel" (meaning "or" in Latin).

In order to take into account all of these different indicators, I use regular ex-

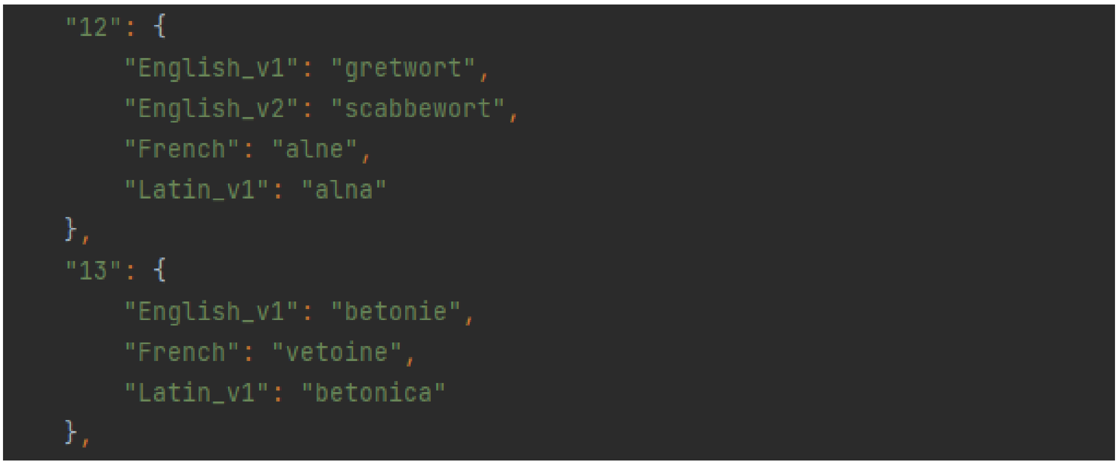
Glossary Entries
Filex arboratica / vel polipodium /: gall. feugerole, angl. everfe[{r{]n
Alga: gall. - angl. sewor
Sambucus: angl. elren
Arsenico[{n{]: angl. orpiment
Assuetus: ang.
Codia: angl. popyesheved oþer blosme
Cissos malus: ang. blac ivi
Alna: gall. alne, angl. gretwort vel scabbewort

Table 13: Different types of entries and annotations in the Trilingual Herb Glossary.

pressions to retrieve the information from the glossary entries. Then, I create a JSON file, where the names in the outer level of the JSON object correspond to the herb ids, and the values correspond to the versions of the herb entity in the given languages. Figure 13 illustrates how the glossary entries are structured in the JSON file.

2.4.2 Herb detection in the CEEM and the potential use of the Trilingual Herb Glossary in the code-switching detection task

As a side task, I decided to search how many of the glossary entries appear in the CEEM texts. I use a glossary lookup method on the token nodes of the XML files, by lower-casing the input tokens to match the already lower-cased glossary entries. If there is a match, I annotate the token with additional information regarding the the herb id number, and the given language. The results of the herb detection side task are presented in Table 14. Despite the strict token-level glossary lookup approach, it is evident that a significant amount of Latin herbs is detected. Such a finding, strengthens my hypothesis regarding the high number of Latin intra-sentential code-switches included in the CEEM texts. My final conclusion



```
    "12": {
        "English_v1": "gretwort",
        "English_v2": "scabbewort",
        "French": "alne",
        "Latin_v1": "alna"
    },
    "13": {
        "English_v1": "betonie",
        "French": "vetoine",
        "Latin_v1": "betonica"
    },
```

Figure 13: Example of entries in the Trilingual Herb Glossary after structured in JSON format.

is that the glossary's Latin entries can be used to increase the Latin data and, subsequently, benefit the task of intra-sentential code-switching detection. This idea will be described in detail in Chapter 3.2.6.

Number of detected herbs	
Latin	340
Middle English	1146
Anglo-Norman	117
Total	1603

Table 14: Results of the token-level herb detection in the Corpus of Early English Medical Writing

2.5 Expected challenges of code-switching detection in the CEEM texts

Having gained a more comprehensive picture of the the features and peculiarities of the CEEM language, it is important to discuss the challenges that these features may trigger in the task of code-switching detection. Code-switching detection is a NLP task accompanied by significant challenges corresponding to both the nature of the detection task itself, and the inherent features of the data. As Martínez

[2020] explains, the identification of code-switches strongly depends on the precise word-level language identification. He proceeds by noting that language identification models tend to perform well on the document, paragraph, and sentence level, but they perform poorly on the word level. Additionally, he stresses how expensive the task of performing word-level language identification is, especially in large corpora [Martínez, 2020, p. 8]. Furthermore, I expect the task of code-switching detection to bring about some additional challenges resulting from the nature of the CEEM data.

As proved in this Chapter, there is a great heterogeneity in the medical writings of the CEEM. This heterogeneity is due to the language shift, the various genres, and authors, as well as the different types of readers to whom the knowledge is addressed [Pahta and Taavitsainen, 2011, p. 9]. Subsequently, automatically processing and systematizing such heterogeneous texts can be demanding. Additionally, when dealing with languages that are closely related as Latin and English, there are many words that are shared between the two vocabularies. For example, the preposition "in" appears both in Latin and English. Usually, the only way to distinguish the language of such words is by looking at the language of their context words. Otherwise, such words cannot be easily attributed with a language label. Another challenge is that the CEEM corpus contains some instances of Ancient Greek and Anglo-Norman words that may interfere with the task of code-switching detection and drop the accuracy of the predictions. Another very common characteristic of the language of CEEM that may complicate the code-switching detetcion task is that many words initially labeled as Latin are actually latinized versions of Greek words. The question that is raised here is how whether such words will be identified as Latin, or as Greek, and subsequently whether they are considered as code-switches or as simple loans. As I mentioned in Chapter 1.3, I will be considering such words as Latin. Ultimately, code-switching is particularly challenging when applied to languages and text genres with a lack of NLP resources, given that there are no code-switching corpora and trained models available for such low-resource languages [Martínez, 2020, p. 2-3].

3 Detecting code-switches in Latin within the Early English Medical Writing Corpus

3.1 Evaluating language Models for language identification on the sentence and token level

My initial idea was to perform the task of code-switching detection in the CEEM texts by applying language models both for the detection of inter- and intra-sentential code-switches. However, after investigating the literature Volk et al. [2022]; Volk and Clematide [2014]; Zhang et al. [2018], I realized that the language models' performance drops significantly depending on the sequence length. Therefore, I first evaluated the language models' performance on sequences from CEEM that I manually labeled as English and Latin.

For this task I compared two language models, following the workflow described in Volk et al. [2022]. The first one is an implementation of the n-gram language model developed by Lenz Furrer and Samuel Läubli, from the University of Zurich. I will be calling it FurL, as it was established in Volk et al. [2022]. FurL is trained separately for each language, and it extracts the corresponding character n-grams per language. When applied for language identification it decides the language label of the given sequence based on the perplexity values of the two possible languages. For this task, I trained FurL on the CEEM data, namely on English and Latin sentences that I manually extracted from the corpus. The second model that I used for comparison is a popular off-the-shelf language identification model called LangID by Lui and Baldwin [2012]. LangID is pre-trained and currently supports 97 languages including English and Latin, without requiring complex configuration. However, despite its practicality, it is questionable how well such a model will perform on Middle English and Medieval Latin data.

In order to gather a representative sample of English and Latin sequences to test the models, I made sure I included data from all three sub-corpora and from various texts. I soon realised that the Latin sequences were shorter in length than the English sequences, mainly because most of them constitute intra-sentential code-switches. Therefore, I decided to gather double the number of Latin sequences such that the comparison of the models' performance is fair. I ended up with 102 sequences of English, and 218 sequences of Latin. The total number of characters of the English sequences is 12,920, and the total number of characters of the Latin sequences is 12,317 characters.

3.1.1 Evaluating the language models on sequences of different lengths

As mentioned, the language models' performance strongly depends on the length of the input sequence. Therefore, I decided to assess the performance of both models on sequences of various lengths and examine how accurate the prediction of the models is. The first group of sequences, on which the models' performance will be tested, is going to be the original sequences, namely, sequences with various lengths as they appear in the CEEM texts. The second group of sequences is going to be sequences with certain length that I generated by truncating the original sequences. More specifically, the first subgroup is going to contain sequences that have a length of approximately 40 characters (±5), the second subgroup is going to contain sequences that have a length of approximately 20 characters (±5), and lastly, the third subgroup is going to contain sequences that have a length of approximately 10 characters (±5).

Initially, I split the original length sequences that I manually collected for each language into training and testing data using k-fold, with k=5. The training data are only used to train FurL since LangID is already trained. Afterwards, I generate additional testing data for each language and fold, by truncating the original testing data to the given length. The workflow that I use for truncating the original sentences to a length n is the following: If the original sequence has a length of $n \pm 5$, then I preserve it in without truncating it, because its original length is not significantly different from the requested length. Otherwise, if the original sequence is longer than the requested length, namely more than 5 characters longer, then it is truncated to the n^{th} character. The goal of this method is to yield precise results regarding the models' performance for each length category.

Because the sequences per language are imbalanced within the different length and fold settings, I decided to compute the balanced accuracy, as seen in Equation 1, in order to deal with the disproportion of data between the two languages.

$$Balanced\ Accuracy = \frac{Sensitivity + Specificity}{2} \tag{1}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{2}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3}$$

When looking in comparison the balanced accuracies of the two models as they are depicted in Tables 15 - 18, it becomes obvious that the performance of FurL is better than LangID's. MBA stands for mean balanced accuracy and SD stands

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	MBA	SD
FurL	1	1	1	1	1	1	0
LangID	0.760	0.736	0.759	0.767	0.719	0.748	0.0179

Table 15: Balanced Accuracies of FurL and LangID on original CEEM sequences of **various lengths**.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	MBA	SD
FurL	1	1	1	1	1	1	0
LangID	0.774	0.814	0.832	0.797	0.752	0.794	0.028

Table 16: Balanced Accuracies of FurL and LangID on truncated CEEM sequences of approximately **40 characters**.

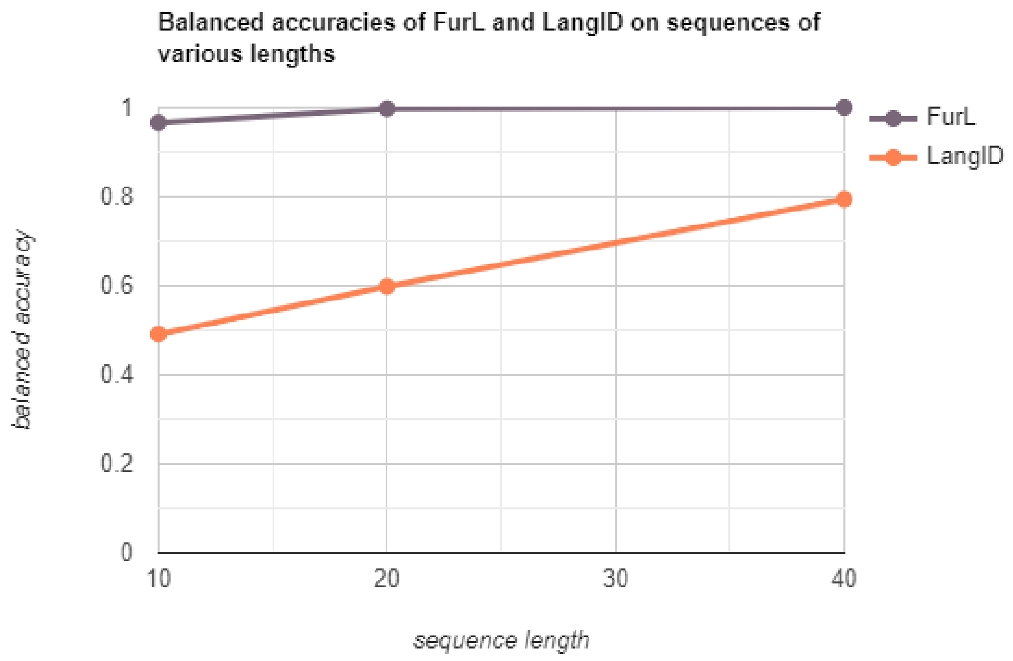


Figure 14: Performance of FurL and LangID on sequences of various lengths.

for standard deviation.

	Fold 1	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**MBA**	**SD**
FurL	1	0.986	1	1	1	0.997	0.005
LangID	0.630	0.601	0.612	0.668	0.482	0.598	0.062

Table 17: Balanced Accuracies of FurL and LangID on truncated CEEM sequences of approximately **20 characters**.

	Fold 1	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**MBA**	**SD**
FurL	0.964	0.929	1	1	0.938	0.966	0.029
LangID	0.487	0.463	0.509	0.509	0.484	0.491	0.017

Table 18: Balanced Accuracies of FurL and LangID on truncated CEEM sequences of approximately **10 characters**.

	Gold LA	**Gold EN**
Predicted LA	218	0
Predicted EN	0	102

Table 19: **FurL's** prediction on original CEEM sequences of **various lengths**.

	Gold LA	**Gold EN**
Predicted LA	115	0
Predicted EN	0	86

Table 20: **FurL's** prediction on truncated CEEM sequences of approximately **40 characters**.

	Gold LA	**Gold EN**
Predicted LA	183	0
Predicted EN	1	101

Table 21: **FurL's** prediction on truncated CEEM sequences of approximately **20 characters**.

	Gold LA	**Gold EN**
Predicted LA	214	5
Predicted EN	4	97

Table 22: **FurL's** prediction on truncated CEEM sequences of approximately **10 characters**.

	Gold LA	**Gold EN**
Predicted LA	119	5
Predicted EN	99	97

Table 23: **LangID's** prediction on original CEEM sequences of **various lengths**.

	Gold LA	**Gold EN**
Predicted LA	80	9
Predicted EN	35	77

Table 24: **LangID's** prediction on truncated CEEM sequences of approximately **40 characters**.

	Gold LA	**Gold EN**
Predicted LA	53	9
Predicted EN	131	92

Table 25: **LangID's** prediction on truncated CEEM sequences of approximately **20 characters**.

	Gold LA	**Gold EN**
Predicted LA	11	7
Predicted EN	207	95

Table 26: **LangID's** prediction on truncated CEEM sequences of approximately **10 characters**.

The superiority of FurL's performance is evident both on the level of the different length settings, and on the level of the folds. FurL's mean balanced accuracy is significantly high on all four length settings with the highest value being 1 for the original length (refer to Table 15) and for the length setting of 40 characters (refer to Table 16). The mean balanced accuracy is slightly decreasing in the setting of 20 characters (refer to Table 17) and 10 characters (refer to Table 18), reaching the values of 0.997 and 0.966 respectively. FurL's high performance is also evident in the respective prediction Tables 19 - 22. When tested on the original sequences (refer to Table 19) and the 40 characters-long sequences (refer to Table 20), FurL successfully identified all Latin and English sequences. When FurL was tested on shorter sequences of 20 characters, it only misclassified 1 Latin sequence as English. Finally, when tested on short sequences of 10 characters, it misclassified 4 Latin sequences as English, and 5 English sequences as Latin, which is approximately 2% of the total data.

LangID's mean balanced accuracy is significantly lower than FurL's for the original sequences, namely, 0.748 (refer to Table 15). When tested exclusively on longer sequences of 40 characters (refer to Table 16), LangID's performance is improved and reaches a mean balanced accuracy value of 0.794. However, there is a dramatic drop in performance when applying the model to sequences of 20 characters (refer to Table 17). The mean balanced accuracy drops to 0.598 and it is decreased even more in the setting of 10 characters (refer to Table 18), reaching the value of 0.491. Moreover, by a first look at LangID's predictions in Tables 23-26, it is obvious that LangID misclassified significantly more sequences compared to FurL. In the original length setting (refer to Table 23), LangID misclassified 99 Latin sequences as English, and 5 English sequences as Latin. This number corresponds to approximately 32% of the data. In the setting of 40 characters (refer to Table 24) LangID misclassified 35 Latin sequences as English and 9 English sequences a Latin. These values correspond to approximately 22% of the data. In the setting of 20 characters (refer to Table 25) the classification is even more deteriorated. More specifically, 50% of the data are misclassified, out of which the 93% are Latin sequences misclassified as English. Finally, when tested on very short sequences of approximately 10 characters (refer to Table 26), LangID misclassified 66% of the data, out of which the 97% is again Latin sequences misclassified as English.

Both LangID and FurL were tested on small datasets with the shortest of the sequences being 5 to 15 characters long. Their performance drops significantly in these settings, and therefore, I assume that when I apply them for language identification on the token level, where the sequences appear to be even shorter than 5 characters, their predictions will be incorrect. The reason why I also include FurL in this conclusion is mostly because of its n-gram nature. Even though an n-gram model like FurL can be easily trained and give meaningful results on the sequence level, when applied to short tokens it can only give a prediction based on the corresponding n-gram probabilities. This means that the model's prediction is heavily dependent on the n-gram counts that appear in the training dataset, which may not always be representative of the n-grams in the whole corpus. This is a problem especially for short tokens like prepositions, which may even be shared among the two languages. A language model like FurL will assign a token with a language label based on the probabilities of its n-grams appearing in each language. This method is meaningless in the case of shared words, and the only solution to resolve similar ambiguities is by taking into account the context. Namely, in the aforementioned case of the preposition "in", which is shared between English and Latin, it would be a good practice to examine the language of the succeeding word and, based on its language label, assign the corresponding language identity to the preposition.

After evaluating the models I draw four conclusions. The first conclusion is the confirmation of my hypothesis that FurL performs better than LangID in all settings, as described in Chapter 1.5. This is due to the fact that FurL was trained on the CEEM data, which boosted its performance, as opposed to LangID. Even though LangID supports both English and Latin, it is not trained on Middle English or Medieval Latin, and therefore cannot recognize the particularities of these two languages as they are represented in the CEEM texts.

My second conclusion is that LangID has a higher tendency to misclassify Latin sequences as English. In my opinion, this tendency is due to the fact that LangID was trained on more English data.

The third conclusion regarding the models' performance is that LangID performs better on longer sequences of data, something which is also described in Volk et al. [2022]. The improved performance of LangID in the 40 characters setting, compared to the original setting, indicates that in the original sequences there is a high number of shorter sequences that causes the model's performance to slightly drop. However, LangID's performance on sequences of less than 20 characters, shows that it cannot be used for language identification on the token level, since tokens' lengths can actually be even less than 20 characters.

This brings me to the fourth and most important conclusion that FurL and LangID can be used for the task of language identification on the level of a whole sequence, but not on the token level. In terms of code-switching, this means that these models can be used for detecting inter-sentential code-switches, which are usually longer sequences, but not for detecting intra-sentential code-switches, since this task requires accurate token level language identification. Eventually, in order to perform accurate language identification on the token level, one needs large word lists for each language. Language models like FurL and LangID can be used as a foundation to generate such lists from the data in question, offering a solution for low-resource contexts. This is the exact approach that is introduced by Volk et al. [2022], and the one that I apply to my own data.

3.2 Lexicon-based Approach

As mentioned in Chapter 1, the work of Volk et al. [2022] constitutes the basic guideline for this thesis' main task of code-switching detection in the CEEM texts. Volk et al. [2022] follow two distinct approaches for detecting the code-switches in the Bullinger Corpus. For the sentence-level language identification, which corresponds to the category of inter-sentential code-switching detection, they use the language models that I just described, namely FurL and LangID. For the token-level language identification, namely the detection of intra-sentential code-switches, they use a lexicon that they generate from data of the same corpus. In brief, their workflow is as follows: Firstly, using the agreement between the two language models, they extract Latin and German sentences from the corpus. Then, they create two lexica, one for each language, based on the sentences they collected. Afterwards, they proceed with filtering these two lexica. Subsequently, they perform word-based language identification and assign the words with a language label, based on the occurrences of the words in the filtered lexica. Finally, they proceed with "resolving"[5] the words that could not be classified using their context, and they detect intra-sentential code-switches by collecting the spans of the Latin code-switches within German sentences and vice versa. In my approach there are some key differences that I explain in detail in the rest of this Chapter.

3.2.1 Using the language models for inter-sentential code-switching detection

As mentioned, the main language of the CEEM texts is English. Therefore, detecting inter-sentential Latin code-switches in the CEEM texts exclusively means that I will be detecting sentences identified as Latin within the English texts. There are no texts in the CEEM having Latin as the main language, and therefore, I will be performing the code-switching detection task to this single direction.

As described in Chapter 3.1, after evaluating FurL's and LangID's performance on sequences of various lengths, I consulted the approach of Volk et al. [2022],

[5] finding the language label

and decided to use the agreement of the two models on the language prediction to collect English and Latin sentences, based on which I will create the Lexica. The sentences of the corpus are cleared from all punctuation symbols in order to be fed to the models. The only punctuation symbols that are preserved is the tilde “∼” and the equal sign “=”, since they are used for scribal abbreviations (refer to Chapter 2.2.3). I also removed numerals, and tokens that include digits, except from the number 3, which is frequently used in the corpus instead of the letter yogh (refer to Chapter 2.2.3). If a sentence is recognized by both FurL and LangID as Latin, then it is saved in a file with Latin sentences. Accordingly, if a sentence is recognized by both models as English, then it is saved in a file with English sentences. If the two models disagree on the language label of a sentence, then this sentence is skipped.

With this method I identified 138,070 English sentences and 2,721 Latin sentences, which are also considered inter-sentential code-switches given that the texts’ main language is English. FurL and LangID disagreed on 19,728 sentences. I decided to save these sentences in a separate file and, at a later step, detect possible intra-sentential code-switches within them. Based on my hypothesis regarding the number of intra-sentential code-switches in the CEEM texts, and after manually examining the English sentences, I discovered that many of the English sentences include Latin code-switches. This is a promising indication that these sentences can be used at a later step for the task of intra-sentential code-switching detection.

Nevertheless, it is important to note that at this stage the data are not perfectly classified. The English file erroneously contains some instances of Latin sentences. Table 27 presents four example sentences which are identified by the two models as English. The first instance is a sentence written entirely in English, the second one is an English sentence which contains a Latin code-switch indicated in bold letters, the third one is an English sentence consisting of two intra-sentential code-switches in Greek, and the fourth one is an entirely Latin sentence. In this task of sentence-level language identification the first three instances are correctly identified as English sentences. However, the third case constitutes an error.

Likewise, the Latin file contains instances of English sentences with Latin code-switches, which were falsely recognized as entirely Latin. Another type of error that I noticed regarding the Latin file is that it also consists of sentences that are written in a third language like Ancient or Medieval Greek and Anglo-Norman French. Table 28 presents five example sentences identified as Latin. The first instance is a sentence written entirely in Latin, the second one is an English sentence which contains a Latin code-switch indicated in bold letter, the third one is a sentence entirely written in English, the fourth one is a sentence, consisting of only one word, written in Greek, and the fifth sentence is written entirely in Anglo-Norman. In this case, only the first instance is correctly labeled as Latin. The second and third instances should have been included in the English file, and the fourth and fifth instances should have been discarded.

Nevertheless, for the task of lexica generation I will be disregarding such errors, mainly because the majority of examples in the files are correctly classified sentences.

English sentence	"DIslocation is the falling of the Joynts out of their Cavities and proper places into another hindering voluntary Motion"	✓
English sentence with Latin code-switch	"When it haþ boiled ynou3 sette it fro þe fire and late it stande stille without mouyng by þe space of a ***pater noster aue maria*** þat þe litarge of þe lede þat is in it may descende to þe grounde and alsone be it 3ette out softly into anoþer panne þat þe litarge be no3t 3ette out withall and þan moue it with a spature strongly vnto þat it be colded"	✓
English sentence with Greek code-switch	"Theophrastus tells us that ***πιτυς*** differeth from ***πεύκη*** among other things in that it is neither so tall nor so streight nor hath so large a leaf"	✓
Latin sentence	"TIRICATA MAGNA"	✗

Table 27: Sentences recognized as English by both FurL and LangID.

3.2.2 Gathering the lexica

The next step towards intra-sentential code-switching detection is the generation of two lexica, one for each language, which will be used for language identification on the token level. More specifically, I generate the two vocabularies by collecting the types for each language and mapping them to their frequency within the collected sentences. For this specific step, Volk et al. [2022] cleared the data from tokens that contain digits. They also removed punctuation symbols at the beginning and at the end of the tokens, they removed square brackets from within the tokens,

Latin sentence	"Cur moritur homo dum salgia crescit in orto"	✓
English sentence with Latin code-switch	"For according to him ***Semen Viri penetrat in testes fæminæ per uteri Tubas***"	✗
English sentence	"Thus sayth Johannicius"	✗
Greek sentence	"Διδασκαλία"	✗
Anglo-Norman sentence	"E cant la dolour de cheif veint oignies la fontayne de la teste en crois e auxi cum sentes que la dolour sey remue axi remuez le unement"	✗

Table 28: Sentences recognized as Latin by both FurL and LangID.

and they preserved upper and lower casing. My sentences are already cleared from punctuation symbols and tokens with digits, and any other annotation symbol is already filtered out during the earliest pre-processing stage described in Chapter 2.2.3. However, as opposed to Volk et al. [2022], I further normalize the words by applying lower-casing, since I am not dealing with German language for which capitalization may be a strong indicator for language identification.

The size of the two unfiltered vocabularies is presented in Table 29. With this approach Volk et al. [2022] gather 158,663 Latin types and 72,089 German types, whereas I gather 122,304 English types and only 11,045 Latin types. Furthermore, after examining the entries of each vocabulary and their corresponding counts, I noticed that even exclusively Latin function words, like the preposition “ad” appears more frequently in the English vocabulary. It is a fact that both the difference between the sizes of the two vocabularies, and the unbalanced occurrence counts per type may negatively affect the word-level language identification predictions. In my opinion, the reason behind the unbalanced occurrence counts between the two vocabularies for Latin words like the preposition “ad”, is my hypothesis that there is a great number of intra-sentential code-switches within the English sentences. As a result, the English vocabulary consists of numerous Latin types, which are part of intra-sentential Latin code-switches. At this point I realised that it would be a good practice to boost the Latin vocabulary types and their counts before filtering the lexica. Therefore, I decided to increase the Latin data.

	Number of types
English Vocabulary	122,304
Latin Vocabulary	11,045

Table 29: Number of types per vocabulary, before filtering.

3.2.3 Increasing the Latin Data

The approach I will be using for increasing the Latin data should meet certain conditions. The first and most important condition is that the new data should increase the counts of common Latin words, like function words. The second condition is that the data should be as close as possible to the Latin language, of the CEEM texts. This means that the CEEM texts and the new Latin data originate from the same period and have a similar domain. The third condition is that after using the additional data the balance between the two vocabularies will be preserved. In other words, the Latin types should not surpass the English types in number, because the corpus' main language is English. Finally, the fourth condition is that the chosen approach should be time-efficient.

One approach would be to use a whole new corpus. This approach may fulfill the first condition, but it is questionable how close such data can be to the language of CEEM, as most Latin corpora include classical Latin. It is also very probable that the third condition will not be met, as the Latin vocabulary will be increased excessively, and even surpass the English vocabulary. Additionally, this approach is too expensive since it requires the pre-processing of a whole new corpus.

Another approach would be to use off-the-shelf Latin glossaries. However, a Latin glossary would not increase the counts per type in the Latin vocabulary, but it would rather increase the total number of types. Moreover, similarly to the corpus approach, these glossaries are most probably generated from Classical Latin corpora. Regarding the third condition, the off-the-shelf glossary may also excessively increase the length of the Latin vocabulary at the expense of the English vocabulary.

While I was searching for a data that fulfill most of the conditions, I came across a version of the Latin Bible, which was publicly available and in XML format [6].

[6] The Latin Bible can be found here.

This Latin Bible version consists of 534,314 Latin words. It can be useful for the task of increasing the counts of function words in the Latin vocabulary, since such words appear in any domain. The Latin Bible is also a product of the Middle Ages and thus, it is chronologically relevant to the CEEM data. As for the domain, even though the CEEM texts have mainly medical content, they quite often include religious segments in Latin, which may vary from a simple invocation of divine entities, to the reference of entire bible passages within the text. Compared to the previous two approaches this one is also time-efficient. Hence, I chose to use this version of the Latin Bible to boost the counts of the Latin vocabulary.

After incorporating the new Latin data into the Latin vocabulary, I ended up with the number of types and tokens that are presented in Table 30. The number of Latin types is almost five times larger, compared to the counts presented in Table 29. In addition, when manually examining the new vocabulary, the counts of Latin function words are significantly increased. For instance, the preposition “ad” initially occured 196 times in the Latin data, and 420 times in the English data. After increasing the Latin data using the Latin Bible, the occurrence of the preposition was raised to 7037 in the Latin data. This boost in the counts is critical, since based on the frequency of types per language, certain words will be filtered out of the vocabularies.

	Number of types
English Vocabulary	122,304
Latin Vocabulary	50,221

Table 30: Number of types per vocabulary, with Latin Bible, before filtering.

3.2.4 Filtering the Lexica

As mentioned, both vocabularies contain words that belong exclusively to the other language. Thus, it is important to filter them before using them for the task of word-level language identification. The method I use to filter the two lexica is similar to the one introduced by Volk et al. [2022]. The authors filter the Latin lexicon by keeping all types that do not appear in the German lexicon. As for the types that also appear in the German lexicon, they only keep those with at least 10 times larger occurrence frequency than the frequency in the German vocabulary. For the German vocabulary, they follow the same approach with the only difference that they keep the types that occurred at least 5 times more than in the Latin vocabulary. They explain that the difference between the factors 10 and 5 reflects the difference in the overall token counts between the two languages.

In the case of CEEM, I also preserve those types in each vocabulary that do not appear in the other vocabulary. Regarding the types that are shared among the two vocabularies, for the Latin dictionary I keep those that have at least 3 times higher occurrence frequency than the frequency in the English vocabulary. For the English vocabulary, I keep those types that occur at least 11 times more than in the Latin vocabulary. Table 31 depicts the new lengths of the lexica after filtering. The high type count in the two filtered vocabularies serves also as a proof of the spelling variation that I expected to encounter while processing the CEEM texts.

	Number of types
English Vocabulary	116,904
Latin Vocabulary	45,391

Table 31: Number of types per vocabulary, with Latin Bible, after filtering.

3.2.5 Using the lexica for intra-sentential code-switching detection in the English sentences

After generating the two filtered vocabularies, I proceed to the task of word-level language identification with the aim of detecting intra-sentential code-switches. More specifically, if a word occurs in the Latin or English vocabulary it is assigned with the corresponding language label. If the word does not occur in any of the two vocabularies it is labeled as unknown. Volk et al. [2022] search for Latin code-switches inside the sentences labeled as German by the language models, and for German code-switches inside the sentences labeled as Latin. On the contrary, because the main language of the texts in the CEEM is English, I will be detecting intra-sentential code-switches in only one direction, namely the Latin code-switches inside English sentences. I will therefore be searching for the code-switches inside the sentences labeled as English, which were presented in Chapter 3.2.1.

In Volk et al. [2022], a code-switching span must have at least the length of two tokens. I apply the same method on my own data. The general approach is the following: For a given sentence, I first label each token as Latin, English, or unknown, while keeping track of the tokens' indices. Then, I search for one or more spans of at least two adjacent Latin words over this whole sequence of labels. Finally, by keeping track of the indices that represent the start and end of the code-switch, I can detect the intra-sentential code-switches inside the sentences. It is important to mention that these label sequences may also include the unknown labels.

Before testing out different techniques to resolve the unknown labels, I decided to experiment more with this approach, and examine how many code-switches are recognized without resolving the unknown tokens. The results [7] are depicted in Table 32. The total number of intra-sentential code-switches is lower than I expected. However, even though these numbers may not represent the true intra-sentential code-switching counts, they can be justified for multiple reasons. The first reason is the high number of unknown-label tokens that is still included in the data. Once the unknown tokens are resolved, I expect the code-switches to increase significantly. The second reason is the fact that I defined the minimum length of the code-switching span to be 2 tokens. This span is promising for precisely detecting code-switches, but based on my hypothesis that a lot of code-switches consist of a single word, this minimum span may also be cancelling a great number of code-switching instances.

Number of intra-sentential code-switches	844
Number of sentences with at least one intra-sentential code-switch	698

Table 32: Intra-sentential code-switching detection in English sentences, without resolving unknown words.

3.2.6 Using the trilingual herb glossary for detecting Latin code-switches

In Chapter 2.4, a particular finding in the MEMT sub-corpus was discussed, namely the Trilingual Herb Glossary and its potential use for the task of code-switching detection. I decided to store all the Latin entries of the glossary in a list and use them for the task of intra-sentential code-switching. This list includes 289 Latin herbs, and seemed promising for increasing the code-switching counts.

After increasing the Latin vocabulary with the Latin herbs, I followed the same steps, as before, to detect intra-sentential code-switches using again two tokens as the minimum code-switching span. This method left the results unchangeable [8]. One reason due to which the sole inclusion of the Latin herbs did not benefit

[7] File path: `https://github.com/EiriniVal/thesis/blob/main/old_cs_results/intra_sent_cs_results_in_en_sentences_unresolved_unk.txt`

[8] File path: `https://github.com/EiriniVal/thesis/blob/main/old_cs_results/intra_sent_cs_results_in_en_sentences_unresolved_unk_with_herbs.txt`

the detection of code-switches may be the fact that the Latin glossary includes a small number of herbs, or that the herbs in the CEEM texts are written with various spellings. However, the most significant explanation behind these results is that most of the herbs are expected to appear as single-word code-switches within the English sentences. Therefore, using a minimum span length of at least two tokens, cancels such instances of code-switches. This explanation is again strongly connected to my hypothesis that there are numerous single token intra-sentential code-switches in the corpus. Therefore, the results are clearly meaningful and they raise the question of how significant the change in the code-switching counts will be, if we also consider spans of one Latin token.

Motivated by my hypothesis I updated the method by also considering code-switching spans of one word if and only if this word is a Latin herb. More specifically, whenever a token appears in the list of the Latin herbs, I label it as Latin, and store its index in a list which contains the indices of tokens that are Latin herbs per sentence. Then I proceed to the detection of code-switching spans of a minimum length of two tokens, with the only exception that now, I also consider code-switches with a span of one token, if and only if their label is Latin and their index is included in the Latin herb indices list.

The results [9] of this technique are presented in Table 33. There is an important increase in the number of code-switches in total, but what also looks promising is that the total number of sentences with at least one intra-sentential code-switch is raised from 698 to 922. This means that 224 sentences include a code-switch with a span of one word, which is also a herb from the dictionary. This increase is important if we consider that the glossary has only 289 Latin herb names and thus, proves that there are even more single-token code-switches in the texts.

Number of intra-sentential code-switches	1,099
Number of sentences with at least one intra-sentential code-switch	922

Table 33: Intra-sentential code-switching detection in English sentences, with Latin herbs, without resolving unknown words, but also considering code-switches with length = 1 token, if this token is a herb.

[9] File path: `https://github.com/EiriniVal/thesis/blob/main/old_cs_results/intra_sent_cs_results_in_en_sentences_unresolved_unk_with_herbs_span_update.txt`

3.2.7 Resolving unknown tokens

The next step towards detecting even more code-switches, and completing the code-switching spans detected so far, is to resolve the identity of the tokens that are labeled as unknown. According to the approach of Volk et al. [2022] when an unknown token is surrounded by two tokens of the same language, the token is assigned with the language label of its surroundings. If the unknown token is the first word in the sentence, then it gets the language of the following word. Similarly, if the unknown token is the last word in the sentence, it gets the language of the preceding word.

In my approach, I roughly follow the same rules as described in Volk et al. [2022]. However, in my approach, when an unknown token is surrounded by English words, it is not labeled as English, but it is left unresolved. The reason is based on my hypothesis regarding the large number of single-token code-switches, and on the findings presented in Table 33, that indicate an important increase when considering single tokens as code-switches. In my opinion, it is probable that an unknown word between two English words is a Latin code-switch, for instance a herb name, a body part name etc. Consequently, labeling unknown words in such context as English is risky and it is a good practice to avoid it. Volk et al. [2022] also use punctuation marks like the comma and the parenthesis for resolving the unknown words' labels. In my approach, such information is ignored, since I search for code-switches on the cleared from punctuation sentences that were initially fed into the language models.

Table 34 illustrates the new counts of code-switches [10]. Both the total number of code-switches and the number of sentences with at least one code-switch have been increased. The total number of code-switches is increased to 1,372 instances. This increase shows the importance of the task of unknown-label resolution. However, these counts should not just be interpreted as an addition of 273 new code-switches over the old number, as this is not the only goal of the code-switching detection task. The hidden advantage of this approach, that is not reflected on the counts, is the fact that not only more code-switches were detected, but also that many of the previously detected spans are now completed. This implies that there are also instances of code-switching spans that merged together because they were separated by an unknown word, which is now resolved as Latin. Therefore, instead of them being counted as two code-switches with incomplete spans, now they are counted as one code-switch with a complete span.

To complete the detection task, I applied the same method for detecting intra-

[10] File path: `https://github.com/EiriniVal/thesis/blob/main/intra_sent_cs_results_in_en_sentences.txt`

Number of intra-sentential code-switches	1,372
Number of sentences with at least one intra-sentential code-switch	1,182

Table 34: Intra-sentential code-switching detection in English sentences after resolving unknown tokens.

sentential code-switches in those sentences with an undetermined language label due to the disagreement between FurL and LangID. Information regarding the counts of intra-sentential code-switches within the unlabeled sentences [11] is included in Table 35. The code-switching counts are significantly high in the unlabeled sentences. However, I expect that this is also due to the fact that there are many Latin sentences in the unlabeled sentences file that could not be classified as Latin because of the disagreement of the two language models.

Number of intra-sentential code-switches	1,423
Number of sentences with at least one intra-sentential code-switch	1,275

Table 35: Intra-sentential code-switching detection in unlabeled sentences after resolving unknown tokens.

	Intra-sentential code-switches	**Sentences with intra-sentential code-switch(es)**
English sentences	1,372	1,182
Unlabeled sentences	1,423	1,275
Total	2,795	2,457

Table 36: Intra-sentential code-switching detection results.

3.2.8 Evaluation of the code-switching detection results

The final counts of the code-switching detection task within the Corpus of Early English Medical Writing are illustrated in Table 37. With the adapted approach of

[11] File path: https://github.com/EiriniVal/thesis/blob/main/intra_sent_cs_results_in_unk_sentences.txt

Volk et al. [2022], I detect 2,457 sentences with intra-sentential code-switches, and 2,721 inter-sentential code-switches [12]. The total number of code-switches detected in the corpus is 5,178. Nevertheless, in order to investigate how these numbers correspond to the real code-switches in the corpus, I perform a manual evaluation to a sample of the data to measure the precision (Equation 4) and recall (Equation 5) values for both inter-sentential and intra-sentential code-switches. In general, precision is going to measure how accurate the detected code-switches are, whereas recall is going to measure how many of the real code-switches are actually detected.

Intra-sentential	2,457
Inter-sentential	2,721
Total	5,178

Table 37: Final code-switching counts.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{4}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{5}$$

Regarding each type of code-switching, the precision value corresponding to inter-sentential code-switches is going to represent how many of the detected inter-sentential code-switches in a sample are correctly detected as sentences written entirely in Latin, while the precision value corresponding to intra-sentential code-switches represents how many of the detected sentences with intra-sentential code-switches actually include Latin code-switches. In the latter case of intra-sentential precision, I manually evaluate each sentence within a sample of sentences with intra-sentential code-switches using three labels, namely “wrong”, “partial”, “total”. The label “wrong” corresponds to sentences, which were erroneously detected as sentences with a Latin intra-sentential code-switch, meaning that they either did not contain a code-switch, or they contained a code-switch in a language other than Latin. The label “partial” corresponds to sentences which were correctly detected as sentences with Latin intra-sentential code-switches, but either the span of the detected code-switches is incomplete, or wrong, or there were more code-switching entities that were not detected. Finally, the label “total” is given to sentences where the Latin code-switches were precisely detected. It should be noted that the sentences in the sample that are entirely written in Latin (inter-

[12] The total number of Latin sentences as they were detected in Chapter 3.2.1

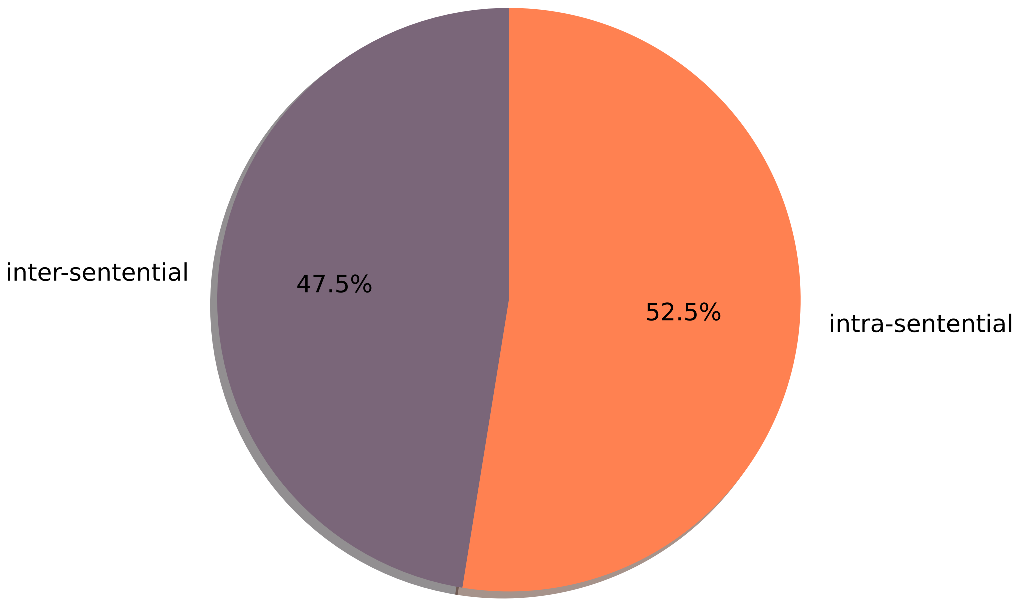


Figure 15: Percentages of the detected code-switches.

sentential code-switches) get the label "partial" if the code-switching span does not correspond to the whole length of the sentence, and the label "total" if their entire span is matched. The reason why I choose this approach is because the methods used for the detection of the two types of code-switching are distinct and my goal is to first evaluate them independently. In the case of intra-sentential code-switching precision, I also follow a loose approach by considering both "partial" and "total" instances as True Positives, mainly because I want to evaluate the task of code-switching detection in the corpus from a wider perspective.

Respectively, the recall value corresponding to inter-sentential code-switches is going to measure how many of some randomly drawn sentences from the corpus, are actually detected as inter-sentential. Ultimately, the intra-sentential code-switching recall value is going to be measured by drawing a sample of sentences, manually detecting sentences with intra-sentential code-switches and searching whether those sentences are detected by the lexicon-based approach. Similarly to the intra-sentential precision, I do not focus on the correctness of the span, but rather on whether the manually detected intra-sentential instances are also detected automatically. Tables 38 and 39 demonstrate selected examples of automatically detected intra-sentential and inter-sentential code-switches, along with their truth labels.

Examples of automatically detected intra-sentential code-switches	
Anno Salutis	TP
E si a vous seyt venu par maladie u par autre survenue pernez le blanche de oyf e le bates ben en une nuvelle vessel e lessez reposer	FP
Succi absinthii pontici majoris three ounces	TP
Though the soule reasonable be made perfect in cunning and vertues as it is sayde in Liber de Anima it is most perfect and most may conteine in the manner of a Circle touching the kindly vertues	TP
Observations sur la Structure des Parties de la Génération de la Femme par Mr Portal	FP
E pus culiez parmy un drap e metez a boystez	FP

Table 38: Automatically detected sentences with intra-sentential code-switches. The code-switching spans are indicated in purple.

For measuring the intra-sentential precision, I used a randomly drawn sample of 150 sentences in total, consisting of 75 sentences from those labeled as English by the two language models, and 75 sentences from those that remain unlabeled by the two models. I did not use sentences labeled as Latin because they constitute inter-sentential code-switches. As for the inter-sentential precision, I used 150 sentences drawn from the whole corpus. I measured recall by doubling the sample size to 300 sentences. For intra-sentential recall, I used 150 randomly drawn English labeled sentences, and 150 randomly drawn sentences from those with unknown language label. For inter-sentential recall, I used 300 randomly drawn sentences from the whole corpus. The sample sizes are summarized in Table 40.

Examples of automatically detected inter-sentential code-switches	
Ceræ albæ ij	TP
Quando tengas mas fortuna mira que es como la Luna	FP
Menthæ	TP
Sit breuis aui nullus tibi somnus meridianus	TP
He says In vesica quidem non vidi sunt tamen in cellu losa tela ei circumposta glandulæ conglobatæ quæ rei fidem faciunt	FP
Les Observations Meteorologiques faites dans les differentes Parties de la France dont le President de la Société a lHonneur de pre senter chaque semaine les resultats au Roi	FP

Table 39: Automatically detected inter-sentential code-switching sentences.

Evaluation sampling sizes				
	Precision		**Recall**	
	inter-sentential	**intra-sentential**	**inter-sentential**	**intra-sentential**
English sentences	-	75	-	150
Latin sentences	150	-	-	-
Unlabeled sentences	-	75	-	150
Whole corpus	-	-	300	-

Table 40: The sample sizes used in the evaluation task per code-switching category.

The final precision and recall results are described in Table 41 and 42 respectively. A visual representation is also shown in Figure 16. I will now proceed to interpret the results per code-switching category, while sharing some of the findings of the manual evaluation task.

The precision value for inter-sentential code-switches is 0.83. This demonstrates

Measuring Precision					
	TP		**FP**	**TP+FP**	**Precision**
inter-sentential	124		26	150	0.83
	partial match	**total match**			
intra-sentential	61	74	15	150	0.90
Code-switching Precision:					0.86

Table 41: Precision values of the code-switching detection task.

Measuring Recall				
	TP	**FN**	**TP+FN**	**Recall**
inter-sentential	5	3	8	0.63
intra-sentential	13	50	63	0.20
Code-switching Recall:				0.41

Table 42: Recall values of the code-switching detection task.

that the particular detection task is rather accurate. I manually detected 124 correctly labeled inter-sentential sentences, in other words, sentences entirely in Latin, out of a total of 150 sentences. Out of the 26 False Positive sentences, 16 are English sentences with or without intra-sentential code-switches, 9 sentences are Anglo-Norman, and 1 sentence is written in Old Spanish. The recall value for inter-sentential code-switches is 0.63. Out of 300 sentences I manually detected 8 inter-sentential code-switches. Out of these 8 sentences only 5 were automatically detected. The recall value for inter-sentential code-switches may be this low due to the fact that the randomly drawn sample of 300 texts has a very small number of inter-sentential code-switches. Therefore, the result may not be representative of the inter-sentential prediction over the whole corpus. Another reason behind the very low recall value is that the true Latin sentences in the sample are very short in length, with some of them even consisting of only one token. As described in Chapter 3.1, the language models' performance, especially LangID's, is worse for shorter sequences. This means that the shorter the sequence, the less probable for the two models to agree on the sequence's language.

The precision value for intra-sentential code-switches is 0.90. I manually de-

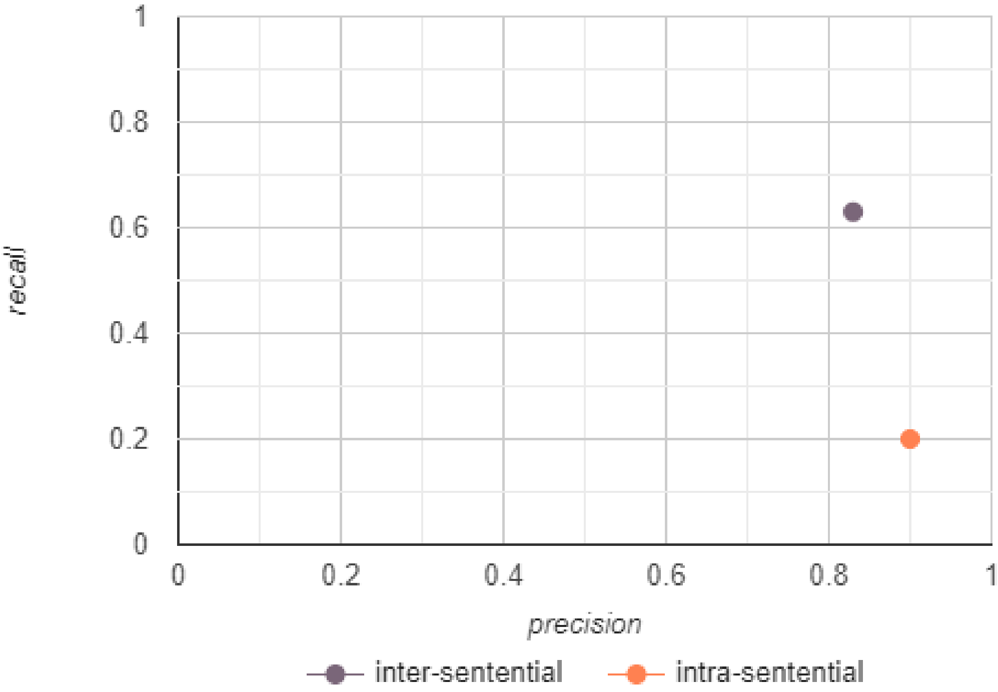


Figure 16: Plot of precision and recall per code-switching category.

tected 61 partial matches and 74 total matches. As mentioned, both the partial and total matches are considered as True Positives. The recall value of the automatic intra-sentential code-switching detection task is 0.20. This means that only a very small proportion of the real sentences with intra-sentential code-switching is automatically detected. Out of the 63 sentences with intra-sentential code-switching, only 13 were automatically detected. The reason behind this poor performance is at the same time the confirmation of my hypothesis that most of the intra-sentential code-switches in the Corpus of Early English Medical Writing have a span of single token, or two tokens. To be more specific, out of the 50 False Negative instances, 38 are sentences containing code-switches with a span of a single token or with a span of two tokens and they are composed by herb names and medical terms. As described in Chapter 3.2.6, the lexicon based approach only considers Latin intra-sentential code-switches with a span of at least two tokens, with the exception of herbs included in the herb glossary. However, even code-switches with a span of two tokens seem to frequently be ignored because the words composing them are not present in the Latin vocabulary.

One could argue based on the recall values that the lexicon-based approach may not be ideal for detecting the intra-sentential code-switches in the corpus and that there should not be the restriction of a minimum length of two tokens for intra-sentential code-switches. However, I have two arguments for supporting that such an approach is ideal for this task, at least at this stage of research. First and foremost, the approach of setting a minimum code-switching length of two tokens actually favors the precision of the model, because it is more probable to have a code-switch if at least two adjacent tokens are in Latin. If this restriction is removed, then the more questionable the predictions will be. Therefore, I strongly believe that it is a better practice to sacrifice high counts of questionable code-switching instances, for lower counts of more certain predictions. My second point is that the performance of the model is still not ideal even for intra-sentential code-switches of two tokens, which are actually instances that the lexicon-based approach considers. This is because the tokens composing the code-switches are domain specific and any approach not using a glossary of Latin medical terminology would not be able to easily identify them. Therefore, this is not a method-specific, but a data-specific problem. In the next Chapter, I also discuss whether Machine Learning models are good methods for the task at hand.

3.3 Machine learning methods for the task of code-switching detection in the Corpus of Early English Medical Writing

In this Section, I comment on the use of Machine Learning for the automatic detection of code-switches in the Corpus of Early English Medical Writing, and on the features and architectures that could be used for such a task. My goal is to explain whether a Machine Learning approach can be used for the task at hand and at which point of research. Before drawing my conclusion, I present some interesting approaches described in literature for the task of code-switching detection. At the same time I assess the advantages and disadvantages that such approaches could have for the given task.

The first approach that I will be discussing is the one by Zhang et al. [2018]. Zhang et al. [2018] use CMX, a simple feed forward network with a globally constrained decoder that maps both code-mixed and monolingual text to the accurate language labels. As training data, they generate two million synthetic code-switching sequences to deal with the issue of deficiency in token-level language labels. The language pairs they use are usually composed by English and a non-English language. They also manually annotate existing code-switching data by sampling posts from Google and comments on Youtube. The CMX model maps tokens to language labels in a two-step process. First, the feed forward network predicts a probability distribution over the labels of each token, using features generated at the level of token and at the level of character. More specifically, the character features used as input for the model, are character n-grams with $n = [1, 4]$, and the value representing a specific n-gram is given by a hash function. They also use script information because certain script types are only associated with a specific language and can be important indicators of the language label. Ultimately, they use Lexicon features, which are generated by a large vocabulary of six million entries that includes the language distributions of words. These Lexicon features include three sub-features, namely the language distribution, an active language feature vector indicating the languages in which a word appears, and the singletons, meaning that if a token is exclusive for a single language, a one-hot vector with the 1 in the dimension of the respective language will be used. In case a word does not match an entry at the lexicon, then the word is matched to languages based on its prefix. In the second step, the decoder decides which sequence of labels is more suitable for the whole input sequence of tokens. They state that using a simple greedy approach and picking the language label with the highest probability per token, will result in predicting too many languages per sentence. This is why they use the global constraint that only monolingual outputs or code-switching outputs of a fixed set of language pairs are permitted in a sentence. After evaluating their model, they claim that their model outperforms other code-mixing and monolingual models. The advantage of the model suggested by Zhang et al. [2018], is that it has a very simple architecture and it takes advantage of features that are critical for any code-switching detection task, considering the level of character, token and the whole sequence as a single unit. They also deal with the lack of code-switching training data by creating synthetic code-switching sequences. This is a very interesting approach that could also be beneficial for language pairs like Middle English and Latin. However, part of their training data is also real-world code-mixing text that comes from social media. When working with historical languages like Middle English and Latin there are either no code-switching corpora or the existing linguistic resources are limited. Subsequently, the task of constructing a large vocabulary from other monolingual corpora in order to extract lexicon features would be even more challenging for the task of code-switching detection in the CEEM, and it would not guarantee the precise language identification of domain specific words that rarely appear on other corpora.

Another approach used for the prediction of word-level language labels in social media texts, is the one by Jaech et al. [2016]. This model was used in the shared task of EMNLP 2016 Second Workshop on Computational Approaches to Code-Switching. They use C2V2L, a hierarchical Neural Network model that predicts the language labels per word by learning character and contextualized word-level embeddings. Their model consists of two parts. In the first part, they generate the embeddings for each word using a Convolutional Neural Network. More specifically, each word is fed to the model as a sequence of characters and an embedding is generated for each character, resulting to a matrix that represents the word. This matrix is convolved more than one times and finally transformed to the final embedding vector representing the word. In the second part, the generated embeddings are fed into a bidirectional Long Short-Term Memory Recurrent Neural Network that maps them to a language label. Their model is trained on Spanish-English data and achieved a high performance when they evaluated their model on the Spanish-English language pairs.

Compared to the approach of Zhang et al. [2018], this approach is more advantageous since it does not demand feature engineering. However, the architecture is computationally expensive and complicated. Additionally, such model requires thousands of training data with code-switches, which is something that we currently do not own, especially for the language pair Middle English-Latin.

The last model that I will be discussing is the one by Samih et al. [2016], which was also used in the same task of EMNLP 2016, and scored first for the the language pair Modern Standard Arabic-Egyptian and second for the language pair Spanish-English. As Samih et al. [2016] describe, most of the Machine Learning systems used for word-level language identification rely on linguistic resources like glossaries, and on morphological and syntactical information. In contrast, their approach uses pre-trained word embeddings from a large Twitter corpus. What is also unique in their approach is that they do not filter their data by language. The reason behind their decision is the assumption that words from different languages will share different contexts. Thus, the embeddings will provide a sufficient distinction between the languages. Their model gets a sequence of tokens as input. Each word and each character are mapped to word and character embeddings respectively. These embeddings are fed into a word-Long Short-Term Memory and a character-Long Short-Term Memory model respectively with dropout applied on both ends. The representations generated from both LSTM models are concatenated and fed into a Conditional Random Field algorithm that predicts the labels for the whole sequence. It is important to mention that the set of labels generated by this model not only include “lang1” and “lang2”, but also the labels “ambiguous”, “named entity”, “mixed”, “foreign word” and “unknown”. They tested different settings of their approach by also including additional features like character n-grams, morphological indicators like upper-casing and punctuation and shape features. However, even without feature engineering their model achieved very high F1 scores for both language pairs. In my opinion this approach can yield promising results for any language pair, including Middle English-Latin, because it does not require feature engineering, and it is language independent. I also believe that the enriched set of labels is a great idea for producing accurate results that can be used for any study related to code-switching patterns. Labeling the named entities is also crucial, as they usually form a barrier in the code-switching detection task.

Observing collectively the aforementioned approaches, I strongly believe that the most efficient Machine Learning architecture for the task of code-switching detection is Recurrent Neural Networks, because they deal with the sequential and temporal nature of language by transmitting information from the earlier hidden layers to the next layers of the model. More specifically, I agree with the use of either a bidirectional Recurrent Neural Network, or a bidirectional Long Short-Term Memory neural network. The bidirectional nature of the model is crucial for the code-switching detection task, since the label of one word does not only depend on the previous words, but also on the following words. Therefore, also considering the direction from right to left would be advantageous. Ideally, I would map the input sequences to pre-trained word embeddings and character embeddings as in Samih et al. [2016]. If I were to decide on additional features for the task of code-switching detection, I would use character n-grams and part-of-speech information to benefit from the morphological and syntactical information that they carry, which can also indicate whether a code-switch is taking place. The final question is whether a similar Machine Learning approach can be currently used for the task of code-switching detection in the Corpus of Early English Medical Writing. The answer is partially negative. In order to apply such a model, we need a great number of labeled training data, which we initially did not have. However, after using the predictions from the lexicon-based approach, and also by generating synthetic data, or using predictions from other corpora, one could actually yield interesting results and experiment with the Machine Learning architectures at a later stage.

4 Quid futurum sit?

The aim of this thesis has been to perform automatic detection of Latin code-switches within the texts of the Corpus of Early English Medical Writing and to investigate the detected categories, namely the inter-sentential and the intra-sentential code-switches. At the same time, while pre-processing the data, my goal has been to investigate the linguistic profile of the corpus.

The first research question that I address is to investigate which type of code-switching is most common in the Corpus of Early English Medical Writing texts. My hypothesis for the first research question is that intra-sentential code-switches of a short length of one or two tokens are most common in the Corpus of Early English Medical Writing. Originally, the foundations of my hypothesis have been the social and linguistic reality enclosing the texts and described in literature, as well as their medical or scientific nature, which reinforces the assumption that a lot of Latin terms from botany and medicine, whose span consists of one or two tokens, are included in the texts.

The second research question that I address is which method is most suitable for the code-switching detection task in the Corpus of Early English Medical Writing. Subsequently, I examine the performance of language models, and of a lexicon-based approach to determine whether language models can be used for such task, or if a lexicon-based approach is the most suitable solution for the code-switching detection task. My hypothesis for the second research question is that language models will perform poorly for such task. I initially support this hypothesis mostly based on the findings from the existing literature that suggest that the shorter the sequence the less probable it is to be labeled correctly.

After researching the contents and the reality surrounding the corpus' texts, I pre-processed and structured the CEEM texts into XML format. I gathered some insights regarding the size and the particular linguistic features of the three sub-corpora. More specifically, I computed the token and type counts, and investigated the use of Old English alphabet characters, as well as that of scribal abbreviations, roman numerals and modern numerals. The results of both the corpus profiling and the type counts already suggested that there is a great spelling variation in the texts. The next side task that I carried out is the pre-processing of the Trilingual Herb Glossary which is included in the MEMT sub-corpus, and the detection of herbs within the corpus' texts. The glossary's Latin entries were also used as an additional material for enriching the Latin Lexicon created for the intra-sentential code-switching detection task. As a next step, I tackled the task of automatic code-switching detection. First, in order to test my hypothesis regarding the poor performance of language models for such task, I tested the performance of LangID and FurL on sentences of different lengths from the corpus that I manually labeled as English and Latin. My hypothesis was confirmed as I concluded that the performance is deteriorated as the sequences get shorter and that such models cannot be used for the task of code-switching detection. I also concluded that FurL's performance is better than LangID's because it has been trained on the corpus' data. I believe that none of these two models could be used for the automatic detection of code-switches, and especially for the detection of intra-sentential code-switches that have a shorter length. However, I have acknowledged their usefulness on detecting the language of longer sequences, such as inter-sentential code-switches, especially when they are used in combination.

Subsequently, I used the lexicon-based approach by Volk et al. [2022] for the code-switching detection task. To summarise, I first used the agreement of the two language models to identify the language of each sentence in the corpus. The sentences labeled as Latin are automatically considered as inter-sentential code-switches. Using the English and Latin sentences, I generated an English and a Latin vocabulary to detect intra-sentential code-switches with a simple lexicon lookup method. I filtered the lexica in a similar way to Volk et al. [2022]. In order to deal with the difference in size between the two vocabularies, and the very low occurrence counts of the Latin word types, I enriched the Latin vocabulary with data from both the Latin Bible and the Trilingual Herb Glossary. I then resolved the words that could not be labeled as English or Latin by applying similar techniques to the ones used in Volk et al. [2022]. I detected 2,457 sentences with more than one Latin intra-sentential code-switches, and 2,721 sentences that constitute Latin inter-sentential code-switches. As a final step, I manually evaluated the results by measuring the precision and the recall values of the predictions for each code-switching category. The precision of both categories is significantly high. However, the recall values of both categories, and most importantly of the intra-sentential code-switches, indicate that there is a great number of code-switching instances in the Corpus of Early English Medical Writing that has not been detected. Specifically, after investigating the intra-sentential instances that were not automatically detected, I have found that the majority of them are code-switches with a length of one or two tokens and they also constitute Latin medical or botanical terms. Both the high counts of intra-sentential code-switches, as well as the low intra-sentential recall value, suggest that my hypothesis regarding the terminological nature of the code-switches, and the superiority in counts of the intra-sentential category within the corpus, is true.

In the last Section, I gave an outlook on three potential Machine Learning approaches proposed in the literature for the task of code-switching detection and I commented on the advantages and disadvantages they may have. I also came to the conclusion that a bidirectional Recurrent Neural Network model could be advantageous for the prediction of language labels on the token level. However, I argued that such a model could not be used for the sake of this thesis due to the lack of annotated training data.

At this point, I acknowledge that the lexicon-based approach has certain drawbacks. The first drawback is that the generation of the vocabularies depends on the sentence level language identification which is performed using language models. This means that, if the language models perform poorly for at least one of the languages of interest, then the less popular language will have less data, and subsequently its vocabulary will be shorter in size or less accurate and representative of the language. Additionally, if the corpus has significantly more sentences in one language and very few in the second language, then the vocabularies would be unbalanced and the task of intra-sentential code-switching detection would be lacking in performance. In such a scenario, one solution that could benefit the performance of the lexicon-based approach would be to use additional data for enriching the less popular language. Such data, could be off-the-shelf glossaries in the language of interest. This is something that I have also implemented by using the Latin Bible and the Latin herbs from the Trilingual Herb Glossary, which proved to be beneficial. However, the most crucial part when incrementing the data is to always make sure that such glossaries cover the domain and period of the texts of interest. A good experiment for future research on corpora with Medieval Latin texts and code-switches would be to use the Medieval Latin glossary of Du Cange also known as "Glossarium mediae et infimae latinitatis", which is also available in XML format. However, using large vocabularies generated from other corpora is a computationally expensive strategy that does not always guarantee a significant improvement in performance, especially when dealing with texts as the ones in the Corpus of Early English Medical Writing, which contain domain-specific terms that may not be part of the off-the-shelf Latin glossaries.

Another drawback of this approach is that there are still many words like the preposition "in" that cannot be resolved. A solution for this problem would be to use part-of-speech information. Namely, if a word is shared between languages and we know that it is a preposition, then we are almost certain that it can get the language label of the following word. Similarly to the idea of including part-of-speech information, and as a recommendation for future experiment and improvement, the use of RNNTagger [Schmid, 2019] on the Corpus of Early English Medical Writing would be beneficial for the detection of Latin code-switches. RNNTagger currently supports both Latin and Middle English and it has a higher accuracy compared to TreeTagger [Schmid, 1994]. After testing RNNTagger on sentences with Latin code-switches, I found that for some of the Latin tokens it assigns the part-of-speech label "fw-la", which means "foreign word-Latin". This tag would be helpful for decisions regarding the span of intra-sentential code-switches, or for assisting the prediction of the token's language. The part-of-speech information could in general be useful for resolving ambiguities as the ones mentioned with the preposition "in", and the lemma information could be used as a normalized version of the differently spelled words, in order to facilitate their mapping to the vocabulary. The drawback of using RNNTagger is that it is computationally expensive and time-inefficient, especially when applied to such a large corpus. Therefore, I only recommend it as a tool for improving or affirming the code-switching detection predictions, but not as the sole method on which to base the code-switching detection task.

The code-switches detected with the Lexicon-based approach in the Corpus of Early English Medical Writing offer, undoubtedly, a fertile ground for further engagement with the task of automatic code-switching detection in medieval medical texts, which can also be extended to other languages such as Greek and French. Additionally, besides improving the detection with the previously proposed methods, the next level of research could be to further analyze the code-switching instances and categorize them into domains, namely mythology, botany, religion etc.

Regarding the final research question of which method is more suitable for the code-switching detection task in this specific corpus, it is evident that the Lexicon-based approach is superior, compared to the use of language models. Furthermore, it does not require a large number of data, contrary to Machine Learning approaches. It leverages the use of monolingual language models by using their agreement for the detection of inter-sentential code-switches. Moreover, the vocabularies used for the intra-sentential code-switching detection are generated using data from the same corpus. This is particularly important for corpora in low-resource languages that are characterized by a lack of linguistic resources. To conclude, the advantage of the lexicon-based approach is that it is resource-efficient. That is, the corpus is not only the target of the code-switching detection task, but it is also utilized to make the latter possible.

References

Jaech, A., Mulcaire, G., Ostendorf, M., and Smith, N. A. (2016). A neural model for language identification in code-switched tweets. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 60–64, Austin, Texas. Association for Computational Linguistics.

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, Spain (Online). International Committee on Computational Linguistics.

Lui, M. and Baldwin, T. (2012). langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea. Association for Computational Linguistics.

Martínez, V. S. (2020). *Identifying and Modeling Code-Switched Language.* PhD thesis, Columbia University.

Pahta, P. (2012). Code-switching in English of the Middle Ages. In *The Oxford Handbook of the History of English.* Oxford University Press.

Pahta, P. and Taavitsainen, I. (2011). *An interdisciplinary approach to medical writing in Early Modern English*, page 1–8. Studies in English Language. Cambridge University Press.

Piotrowski, M. (2012). *Natural Language Processing for Historical Texts.* Springer International Publishing.

Reimer, S. R. (1998). Manuscript studies: Medieval and early modern. `https://sites.ualberta.ca/~sreimer/ms-course/course/abbrevtn.html`.

Samih, Y., Maharjan, S., Attia, M., Kallmeyer, L., and Solorio, T. (2016). Multilingual code-switching identification via LSTM recurrent neural networks. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 50–59, Austin, Texas. Association for Computational Linguistics.

Schendl, H. and Wright, L., editors (2011). *Code-Switching in Early English.* De Gruyter Mouton, Berlin, Boston.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK.

Schmid, H. (2019). Deep learning-based morphological taggers and lemmatizers for annotating historical texts. In *Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage*, DATeCH2019, page 133–137, New York, NY, USA. Association for Computing Machinery.

Schneider, G. (2020). Spelling normalisation of late modern english: comparison and combination of vard and character-based statistical machine translation. In Kytö, M. and Smitterberg, E., editors, *Late Modern English: novel encounters*, number 214 in Studies in language companion series, pages 243–268. John Benjamins Publishing, Amsterdam.

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*, pages 43–51, Berlin, Germany. Association for Computational Linguistics.

Taavitsainen, I. and Pahta, P., editors (2009). *Studies in English language: Medical and scientific writing in late medieval English.* Studies in English language. Cambridge University Press, Cambridge, England.

Taavitsainen, I. and Pahta, P. (2013). The Corpus of Early English Medical Writing (1375–1800) – a register-specific diachronic corpus for studying the history of scientific writing. `https://varieng.helsinki.fi/series/volumes/14/taavitsainen_pahta`.

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. Association for Computational Linguistics.

Volk, M., Fischer, L., Scheurer, P., Schroffenegger, B. S., Schwitter, R., Ströbel, P., and Suter, B. (2022). Nunc profana tractemus. Detecting code-switching in a large corpus of 16th century letters. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2901–2908, Marseille, France. European Language Resources Association.

Wulff, H. R. (2004). The language of medicine. *Journal of the Royal Society of Medicine*, 97(4):187–188.

Zhang, Y., Riesa, J., Gillick, D., Bakalov, A., Baldridge, J., and Weiss, D. (2018). A fast, compact, accurate model for language identification of codemixed text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language*

Processing, pages 328–337, Brussels, Belgium. Association for Computational Linguistics.