

“Indexing and Analyzing Scriptural References in Early English Printed Sermons”

Abstract

Early printed sermons frequently contain numerous textual and marginal scriptural references, which Arnold Hunt, in *The Art of Hearing*, describes as a “defensive barrier” that preachers erected in print to strengthen their arguments (12). Quantifying the distribution, diversity, granularity, and prominence of preachers’ proof texts offers valuable insights into their doctrinal foci, preferred Bibles, intertextual relationships, and referencing patterns across various historical periods. However, early books often lack indices, and reference identification is further complicated by inconsistencies in formatting and spelling. Using an interdisciplinary text mining approach, this paper identifies and analyzes scriptural references extracted from the sermon-related sections of early printed books transcribed by the Text Creation Partnership for the *Early English Books Online* database (EEBO-TCP). This project builds an iterative pipeline for identifying scriptural quotations and paraphrases using custom adaptations of MacBERTh, a large language model pre-trained on historical English corpora, supplemented with a rules-based approach of parsing accompanying citations. The mined references and fine-tuned models are then used to provide an interactive reference index and custom semantic search engines for book titles, marginalia, and six Bibles on a dedicated website (www.earlyenglishprintedsermons.org). This paper further appraises the affordances and limitations of computational methodologies in rectifying and quantifying the errors, representations, and recycling of sources in early sermons.

Introduction

Early printed sermons frequently contain numerous textual and marginal scriptural references, which Arnold Hunt, in *The Art of Hearing*, describes as a “defensive barrier” that preachers erected in print to strengthen their arguments.¹ In addition, such publications may also be laden with a variety of theological, philosophical, and literary ideas from patristic, classical, medieval, and contemporary sources. The preparer of a printed sermon—who may be the preacher himself or an audience member relying upon memory and notes—frequently revised a sermon’s references and style in order to adapt the initially oral performance for dissemination in print.² Given the deliberate decision-making surrounding these intertexts, a quantification of their distribution, diversity, granularity, and prominence promises fascinating macrohistorical insights on preachers’ doctrinal foci, preferred Bibles, linguistic style, and overall intellectual frameworks.

In his detailed analysis of early modern sermons, Hunt draws attention to how one seventeenth century preacher’s habit of providing Latin quotations followed by their English translations was “unusual – and distinctly old-fashioned.”³ Such a characterization naturally raises questions about the extent and proportions of different preaching styles that remain in the existing record of printed sermons. Is it possible to quantify exactly how many sermon collections or published preachers are similarly old-fashioned? What passages from different sources are being quoted most prominently by preachers of various theological inclinations, during distinctive occasions, and across important historical moments? Hunt also notes that certain sermon audiences had very particular and unexpected demands; they desired sermons “salted with learned quotations but limited to relatively simple paraphrases of the familiar Gospel stories.”⁴ Exactly how often do surviving sermons fit these criteria? What is the prominence of the Old Testament (O.T.) references in comparison to those of the New Testament (N.T.) and even the Apocrypha in printed sermons? Can we glean knowledge about preachers’ typological hermeneutics by examining which O.T. passages co-occur with N.T. ones? Are there continuities in referencing patterns across the sometimes nebulous denominational and political divides? The question about the likely version(s) of the Bible featured in each sermon is also significant, as

¹ Arnold Hunt, *The Art of Hearing: English Preachers and Their Audiences, 1590–1640* (Cambridge: Cambridge University Press, 2010), 12.

² *Ibid.*, 146-7.

³ *Ibid.*, 264.

⁴ *Ibid.*, 268.

Thomas Fulton's *The Book of Books* demonstrates how different Biblical texts come with distinctive paratexts that determine their political usage and particular Anglicizations that influence theological interpretations.⁵ We may ask, are there (re)-publications of sermons that quote from certain Bibles, such as the Geneva version, during years that feature momentous historical events in post-Reformation England? These are some of the questions which motivate this project. In an effort to explore the interconnections between source texts in a genre so crucial to the theological, ethical, and political discourses of Renaissance and Early Modern England, this project aims to uncover statistical insights on the structure of scriptural quotations in the sermon-related sections of early printed books transcribed by the Text Creation Partnership for the *Early English Books Online* database, hereafter EEBO-TCP.⁶ This project further appraises the affordances and limitations of computational methodologies to rectify and quantify the errors, representations, and recycling of such a distinctive form of generationally inherited, shared knowledge in these sermons.

Early books are rarely accompanied with convenient indices, and indeed only 73 books in EEPS' corpus (less than 1.3 percent) have tables or indices of scriptural references. More than that, references in early books appear in such a vast variety of formatting and spellings within the span of over two centuries that simply searching for exact matches of known keywords or passages is wholly insufficient. Illegible words and characters, the complete absence of quotation marks, and variable title and authorial abbreviations likewise complicate reference detection. Therefore, this paper presents a systematic methodology for mining and analyzing scriptural text reuse in the body texts and marginalia of sermons in the following order: identifying and extracting sermons from EEBO-TCP, annotating sermons with linguistic information, extracting citations using heuristics, and identifying quotations and paraphrases of verses from six different Bibles. The Bibles considered in this project are the Authorized King James Version of 1611 (hereafter AKJV), Geneva Bible of 1599, a combination of the available plain-text transcriptions of the original and modernized Douay-Rheims version, the Latin Vulgate, William Tyndale's New Testament, and John Wycliffe's version of the Pentateuch and Gospels.⁷ This project focuses on three textual dimensions of analysis with relation to scriptural quotations: Latinity, typographical emphasis, and literality. Specifically, I aim to identify how scriptural quotations and paraphrases from these Bibles are distributed in the text and marginalia relative to each other and to spans of texts that are in a foreign language or typographically emphasized using italicization.⁸ Additional metrics are diversity, evenness, and prominence of text reuse for each Bible part, book, chapter, and verse. Diversity and evenness can be measured

⁵ Thomas Fulton, *The Book of Books: Biblical Interpretation, Literary Culture, and the Political Imagination from Erasmus to Milton* (Philadelphia: University of Pennsylvania Press, 2021), 3-10.

⁶ Text Creation Partnership. "Early English Books Online (EEBO) TCP – Text Creation Partnership." *Text Creation Partnership*, Accessed June 5, 2025. <https://textcreationpartnership.org/tcp-texts/eebo-tcp-early-english-books-online/>.

⁷ I used the JSON files for each book of the AKJV version from this repository: Arul John. *Aruljohn/Bible-Kjv-1611*. 24 Dec. 2024. *GitHub*, <https://github.com/aruljohn/Bible-kjv-1611>. The AKJV contains 36,702 verses. For the Geneva version, see *Geneva Bible 1599*. <https://ebible.org/find/details.php?id=enggnv>; the Geneva version contains 31,090 verses. The Douay-Rheims version is a special case because there is only a partial transcription of its original text. There are only 14,736 verses available online in plain text from the microfilm scans of the original Douay Rheims version (hereafter ODRV), including the entire New Testament of 1528 but only a few books of the Old Testament of 1610. I extracted these available verses by directly scraping a website dedicated to the ODRV; see *Original Douay Rheims Bible (1582 & 1610)*. <https://originaldouayrheims.com/home>. The remaining 21,065 verses of the Douay-Rheims version come from the American Edition of 1899; see *Douay-Rheims 1899*. <https://ebible.org/find/details.php?id=engDRA>. The version of the Vulgate I use is the 1880 Glossa Ordinaria Migne edition of the 1598 Clementine Vulgate, which contains 35,808 verses; see *Bibbia Vulgata Clementina Na 1598*. <https://ebible.org/find/details.php?id=latVUC>. The Tyndale New Testament contains 7,954 verses; see *Tyndale New Testament*. <https://ebible.org/find/details.php?id=engtnt>. The Wycliffe Bible of c.1395 contains 9,622 verses; see *Wycliffe Bible*. <https://ebible.org/find/details.php?id=engWycliffe>. I extracted and organized these Bibles into CSV files, which I share in my GitHub repository: <https://github.com/amycweng/Early-Modern-Sermons/tree/main/assets/Bibles>. Note: the URL prefix for all code and data files mentioned in the notes is <https://github.com/amycweng/Early-Modern-Sermons/tree/main>.

⁸ Notes are found under the 'NOTE' element of the transcriptions, and the 'HI' element tag demarcates the boundaries of typographically emphasized text. Foreign spans of text fall under two categories: (1) gaps of non-transcribed content (indicated by a 'GAP' element with the description '{ in non-Latin alphabet }' and (2) phrases of foreign text annotated using specialized software for Early Modern English.

using the metrics that are used for measuring species diversity in biology and citational diversity in bibliometric analyses.⁹ Moreover, what are the most *prominent* references associated with each publication, author, Library of Congress subject heading, publication place, publication year, or historical era? I measure prominence using the Outgoing Relative Citational Prominence (ORCP) metric named by Wahle et al. (2023) in a paper aptly titled “We are Who We Cite.”¹⁰ The ORCP is a percentage calculated by taking the difference between the proportion at which an individual entity references a particular source and the average proportion of that source for all referencing entities in a given group. The EEPS project also divides its corpus into nine historical eras for organization and analysis: pre-Elizabeth, Elizabeth, James I, Charles I, Civil War, Interregnum, Charles II, James II, and William and Mary.¹¹ I name the first era “pre-Elizabeth” rather than each monarch from the mid-fifteenth century to 1557 because the size of that subcorpus is too small to be representative when divided further. However, the classification of works published during boundary years is problematic, especially since March 25 marked the beginning of the new year in England until the mid-seventeenth century, and England continued to follow the Julian calendar until the mid-eighteenth century.¹² Unfortunately, most publication dates do not include months.

In reality, the most important metadata to consider are not the aforementioned ones but the original preacher, the likeliest place of preaching, the year of original preaching, and the names and locations of the listed printer or bookseller, but these require extensive annotation in order to standardize spellings and recognize similar matches, and many of these details are not available for every publication. The author listed in the catalog for a publication is not necessarily the preacher, especially in anthologies. More granular and comprehensive analyses of these sermons in relation to these forms of metadata will only be possible in the future as the EEPS project grows and improves.

Corpus Overview

Complexities have arisen since the beginning because what qualifies as a sermon in print is not straightforward, even though the TCP’s scholars annotated each division in the XML transcriptions with a section name. For instance, over nineteen thousand section divisions in EEPS’ curated corpus of 5,725 publications within EEBO-TCP have labels identifying them as originally oral material, such as a sermon, lecture, or homily. By the term “section,” I am referring to the “DIV” elements in the transcriptions, which range from “DIV1” to “DIV7” such that “DIV2” is a subsection of “DIV1” and so forth. A DIV1 section identified as a “sermon” can contain its own text and a DIV2 subsection named “part,” which I count as hierarchical units respectively named “sermon” and “sermon→part.” There are 26,254 relevant section units, of which 19,640 (74.8 percent) contain at least one sermon-related identifier.¹³ However, a conservative approach of relying only on these identifiers in the TCP transcriptions misses many sermons due to

⁹ Kathleen A. Nolan and Jill E. Callahan, “Beachcomber Biology: The Shannon-Weiner Species Diversity Laboratory Teaching, Volume 27, ed. Michael A. O’Donnell (Proceedings of the 27th Workshop/Conf Laboratory Education [ABLE], 2006), 334-38, <https://www.ableweb.org/biologylabs/wp-content/upl> Chun-Kai Huang et al., “Open Access Research Outputs Receive More Diverse Citations,” *Scientomet* 825-45, <https://doi.org/10.1007/s11192-023-04894-0>.

¹⁰ Jan Philip Wahle et al., “We Are Who We Cite: Bridges of Influence Between Natural Language Proc Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, ed. Houda E (EMNLP 2023, Singapore: Association for Computational Linguistics, 2023), 12896-913, <https://doi.org/10.18653/v1/2023.emnlp-main.797>.

¹¹ Here are the inclusive date ranges and number of sermons per era: pre-Elizabeth (1400-1557; 83 books), James I (1603-1624; 674 books), Charles I (1625-1641; 543 books), Civil War (1642-1649; 617 books), Charles II (1660-1684; 1497 books), James II (1685-1688; 256 books), and William and Mary (1689-1702; 1689-1702; ambiguous date range, such as “1600-1699?”, I arbitrarily choose the first mentioned year so that it matches the date range of the next era).

¹² Sarah Werner, *Studying Early Printed Books, 1450-1800: A Practical Guide* (Wiley, 2019), 84.

¹³ In total, there are 56,193 section units in the entire corpus, so I only extracted 46.7 percent of them. There are 4,259 books with sermon-related section units, and here are the top 10 most frequently occurring units with an obvious sermon-related component: [(‘sermon’, 7965), (‘sermon→part’, 2763), (‘sermons→sermon’, 1769), (‘text→sermon’, 809), (‘sermon→section’, 416), (‘sermon→chapter’, 314), (‘part→sermon’, 308), (‘sermon→application’, 302), (‘part→lecture’, 301), (‘lecture’, 254)].

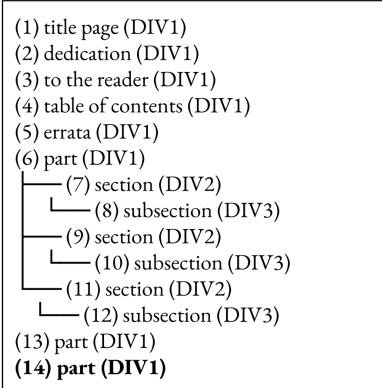


Figure 1: Book Structure for A64635: *Certain Discourses, Viz. of Babylon...* (1659)

transcribers' subjective labeling decisions. Thus, I also use relevant keywords in the subject headings and titles, particularly variants of “sermon” or the past tense of “preach,” to identify relevant publications, which results in over a thousand books which comprise section types like “treatise,” “tract,” “part,” or “discourse.”¹⁴ These variations are unsurprising because religious genres sometimes overlap and resemble each other in a way that, as Hunt says, “almost defies categorization.”¹⁵ Nevertheless, they greatly complicate the procedure of sermon extraction. For example, there is a publication from 1659 that only contains a single relevant section, a sermon from a bishop, in the midst of purely written discourses, a character description of this bishop, and many paratextual materials; I identified this publication for inclusion not by its subject terms or its section identifiers, but rather by the mention of a sermon in its title, and I needed to label that its fourteenth distinct section, a DIV1 section labeled “part,” is the only component I should extract for analysis (see Figure 1).¹⁶

Multi-work, bound volumes or works with a multitude of different section types thus require annotation, so I accordingly labeled over two hundred EEBO-TCP titles, sixteen of which have annotations that are precise to the level of a page range or a subset of a given section type.¹⁷ Among the publications I include in my corpus, special cases of sermon afterlives are poems and quotations (“memorables”) that are adapted directly and expressly from sermons, as well as “honest accounts” and summary heads of sermons. To exclude false positives, I manually review the candidate books that do not contain sermon-related section types. Some transcriptions only contain the title page and colophon, whereas others are bound books within which the sermon(s) mentioned in its title do not survive. The majority of excluded publications are responses to sermons, or preachers' responses to those responses, that usually contain one of the following terms in their titles: remarks, answer, observations, animadversions, discourses, response, commentary, censure, and defense. Others are treatises, poems, mock sermons printed on broadside sheets, dialogues, catalogues, guidebooks to hearing sermons, letters, catechisms, narrative accounts, hymns, petitions, testimonies, and proclamations—all with no direct or clear oral predecessor. I excluded 271 books after examining their contents and 42 foreign-language books identified using the language code within each transcription.¹⁸ Finally, I excluded two books with a publication year after 1702, which is the boundary of the last era in this corpus, the reign of William and Mary.

After sermon identification and section analysis, the next steps are extraction and linguistic adornment.¹⁹ Extraction involves converting relevant sections in the transcriptions into plain text, including all their subsections except tables, indices, and errata lists, as well as retaining the location and boundaries of notes, italicized content, pages (or page images), paragraphs, and sections using corresponding placeholders. Illegible letters and words display respectively as individual dots “•” and bracketed rhombi “⟨ ◊ ⟩”.²⁰ Afterwards, I process each plain text file using the specialized software of MorphAdorner v2.0, a historical English linguistic “adornment” software developed at

¹⁴ There are 1,466 books lacking obvious sermon-related identifiers, but they actually comprise 6,614 relevant units in total. Here are the top 10: [(‘text’, 922), (‘text→part’, 779), (‘text→chapter’, 576), (‘text→thesis’, 373), (‘treatise→chapter’, 350), (‘part’, 289), (‘text→section’, 198), (‘treatise→part’, 183), (‘part→chapter’, 181), (‘biblical_commentary→part’, 164)].

¹⁵ Hunt, *Art of Hearing*, 117.

¹⁶ James Ussher, *Certain Discourses, Viz. of Babylon (Rev. 18. 4.) Being the Present See of Rome (with a Sermon of Bishop Bedels upon the Same Words) of Laying on of Hands (Heb. 6. 2.) to Be an Ordained Ministry, of the Old Form of Words in Ordination, of a Set Form of Prayer: Each Being the Judgment of the Late Arch-Bishop of Armagh, and Primate of Ireland / Published and Enlarged by Nicholas Bernard ... : Unto Which Is Added a Character of Bishop Bedel, and an Answer to Mr. Pierces Fifth Letter Concerning the Late Primate.*, 1659, <http://name.umdl.umich.edu/A64635.0001.001>. These charts are available for each publication in their dedicated pages on the EEPS website; e.g., see www.earlyenglishprintedsermons.org/A64635 for *Certain Discourses*.

¹⁷ See this ‘[lib/dictionaries/sermons_annotations.py](#)’ in my code repository for my annotations. Books that have single, unambiguous section identifiers listed in the ‘wanted_sections’ list (located at the bottom of this Python file) do not require further labeling. Thankfully, most publications fall under these relatively simple cases.

¹⁸ The language code is located in an element named ‘LANGUSAGE.’

¹⁹ See ‘[lib/EEPS_extract.py](#)’ and ‘[lib/EEPS_adorn.py](#)’ in my repository.

²⁰ I do not convert these to standard ASCII characters like asterisks because MorphAdorner recognizes asterisks at the beginning of words to be separate tokens from the word. For example, MorphAdorner turns “*ffection” into “*” and “ffection”, but it keeps “•ffection” as is.

Northwestern University in 2013.²¹ MorphAdorner’s “adornplainemetext” tool (where “eme” is Early Modern English) for plain-text files outputs a plain-text file in which every input token is adorned in tab-separated rows containing its token, standard spelling, part of speech, standard modern spelling, lemma, and binary end-of-sentence (EOS) label.²² Adornment is necessary and useful because it recognizes spans of text in a foreign language like Latin or French, determines whether a period following a letter or number is likely to be the end of a sentence rather than part of an abbreviation or citation, and predicts which tokens are likely to be highly informative parts of speech like proper nouns and cardinal numerals.

The adornments are particularly useful for an ensuing segmentation pipeline that turns paragraphs of text in these books into smaller segments more suitable for processing using LLMs, which in turn convert input passages into fixed-size, dense lists of real numbers, or word embeddings, representing the passage’s semantics in vector space. These embeddings are dense because most of their elements are non-zero, and they are contextual rather than static because they take into account a token’s surrounding context; a token’s embedding changes depending on its adjacent tokens. Each pre-trained LLM has its own tokenizer, which usually applies subword tokenization (breaking words into subwords and unknown words into individual characters) to handle out-of-vocabulary words. This means that spelling regularization is not necessary for this architecture. Modern Information Retrieval (IR) systems often comprise a retriever followed by reranker, which ranks the top-k results returned by the former, and usually both models produce dense, contextual representations for sentences by averaging its individual word embeddings, a strategy known as mean pooling. This means that the embedding for a phrase with a dozen tokens is the same length as that of a passage with a hundred tokens. An important consideration is that this search system can either be asymmetric or symmetric depending on the discrepancy in length between the query and the passages in the corpus that the system retrieves and reranks. The tasks of determining whether a passage contains an actual Biblical quotation and then ranking the most similar verses require a system optimized for symmetric search, since I am assuming that scriptural quotations and their source texts will be mostly comparable in length. EEPS’ procedure for discovering scriptural quotations relies on fine-tuning MacBERTh, a BERT-based model pre-trained on massive historical English corpora by Manjavacas and Fonteyn (2022), using the SBERT architectures developed by Reimers and Gurevych (2019) for efficient sentence-level semantic similarity and clustering tasks.²³ MacBERTh’s tokenizer ignores casing, which is both an advantage and a limitation: it shrinks the input corpus, but the model may also lose some sensitivity to proper nouns. Moreover, the model truncates all input passages to a maximum sequence length of 128 subtokens, and all my versions of MacBERTh, fine-tuned with SBERT, have this same limit.

As such, my next step is to divide each extracted section unit into smaller, discrete segments with lengths that are roughly equivalent to the average length of a Bible verse. For simplicity’s sake, my segmentation algorithm does not use MacBERTh’s tokenizer but rather the count of non-punctuation tokens determined by MorphAdorner. I use the term “segment” rather than “sentence” in this context because I am not merely splitting text by modern sentence delimiters: the period, question mark, or exclamation point. As Julianne Werlin explains, the meaning and composition of the modern “sentence,” a grammatical unit delimited by punctuation, developed gradually in tandem

²¹ Philip R. Burns, “MorphAdorner” (Information Technology (NUIT), Academic Technologies (NUIT), August 1, 2013), <https://morphadorner.northwestern.edu/>. Following MorphAdorner’s example, I use the term “adorn” because, as that project’s author describes, “adornment” is less ambiguous and “harkens back to the medieval sense of manuscript adornment or illumination -- attaching pictures and marginal comments to texts, as the scribal monk at right is doing.” Together, extraction and adornment take a total of 19 hours to complete.

²² MorphAdorner’s part of speech tagset is “NUPOS”; see Philip R. Burns, “MorphAdorner: NUPOS” (Information Technology (NUIT), Academic Technologies (NUIT), August 1, 2013), <https://morphadorner.northwestern.edu/morphadorner/documentation/nupos/>.

²³ Enrique Manjavacas and Lauren Fonteyn, “Adapting vs. Pre-Training Language Models for Historical Languages,” *Journal of Data Mining & Digital Humanities* NLP4DH, no. Digital humanities in languages (June 13, 2022), <https://doi.org/10.46298/jdmdh.9152>; MacBERTh and GysBERT, “MacBERTh,” MacBERTh, accessed May 7, 2025, <https://macberth.netlify.app/>; Nils Reimers and Iryna Gurevych, “Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks” (arXiv, August 27, 2019), <https://doi.org/10.48550/arXiv.1908.10084>;

with the proliferation of print technologies during the Renaissance.²⁴ Thus, Renaissance writers often delimited their sentences with semicolons and conjunctions, the latter of which is a carryover from habits of speech:

Although by 1600 everyone used punctuation to delimit sentences, the distinction between periods and semicolons was much less clear-cut than it is today. The extraordinarily long sentences of Renaissance prose can in part be attributed to the indefiniteness of sentence boundaries. In addition, the heavy use of sentence-initial conjunctions, such as “for,” “and,” or “but,” suggests an attempt to distinguish sentences lexically rather than via punctuation—a conception of the sentence as still partially oral rather than wholly literate....Oral discourse is highly paratactic, marked by a much higher use of “and” than writing, and a great deal of fifteenth- and early sixteenth-century writing, even by relatively well-educated writers, shared this characteristic.²⁵

Hunt describes this period of transformation as an “intricate and long-drawn-out process in which orality became assimilated into a mixed oral/literate culture.”²⁶ Acknowledging the orality of early modern writing is especially important for researching early modern sermons. Despite the fact that preached and printed sermons sometimes differ greatly in content and style as a result of preachers’ interventions to remove “the more obvious traces of oral delivery,” Hunt argues that there was a significant demand for printed versions faithful to the spoken word—ones that “mimicked the plain style and emotional effects of oral preaching.”²⁷ Taking into account these factors, EEPS divides text not only by full-stop punctuation, but also by semicolons, colons, and slashes, which were used particularly with verses. To split up overly long sentences, I am allowing subordinate clauses to be individual clauses as long as they come after a comma and begin with a known conjunction or transition word (see the table of these words below). As follows in the algorithm description below, I also enforce rules that prevent notes, short italicized phrases, parentheticals, and citations from being divided into different segments, and I do not allow any segments to cross section boundaries. Gaps for missing or illegible pages likewise delimit segments. To avoid subdividing any scriptural citations located in the body paragraphs, I do not allow segments to begin with Arabic and Roman numerals if those numerals do not come after a semicolon, question mark, slash, or exclamation point; when encountering one of these numerals, I always add it to the end of the current running segment. Moreover, no segment should end with a numeral when the next segment starts with a known conjunction or transition word after a comma or period.²⁸

²⁴ Julianne Werlin, *Writing at the Origin of Capitalism: Literary Circulation and Social Change in Early Modern England* (Oxford University Press, 2021), 37. Werlin notes that the older meaning of “sentence” is a *sententia*, i.e., an aphorism.

²⁵ *Ibid.*, 38.

²⁶ Hunt, *Art of Hearing*, 59.

²⁷ Hunt, *Art of Hearing*, 162-3.

²⁸ *Relevant Conjunctions and Transition Words*: &, After, Aftir, Alas, Albe, Albeit, Als, Also, Although, Althou, Although, And, Ande, Anone, As, Becaus, Because, Before, Behold, Beholde, Bicause, But, Bycause, Ecce, Ergo, Et, Etiam, Euen, Euēn, Even, Except, Eyther, Ferthermore, Fifthly, Finally, First, Firstely, For, Forasmuch, Forasmuche, Forsothe, Forthermore, Forthly, Fourthly, Further, Furthermore, Hence, How, Howbeit, Ideo, If, Like, Likewise, Lo, Loke, Looke, Lyke, Lykewyse, Marke, Moreouer, Moreover, Nam, Namely, Nay, Naye, Ne, Nec, Neither, Neuer, Neuertheles, Neuerthelesse, Neyther, Nor, Now, Nowe, Or, Post, Quapropter, Quare, Quia, Quoniam, Secondely, Secondly, Sed, Sequitur, Sic, Sicut, Since, Sine, Sith, Sithe, Sithen, Sithence, So, Surely, Surelye, Thā, Than, Thanne, Then, Thenne, Thenē, Therefore, Therfor, Therefore, Thirdly, Though, Thoughe, Thus, Thyrdely, Thyrdly, Til, Tille, Truely, Truly, Unless, Until, Unto, Verely, Vnto, Well, Whan, Whanne, Wheither, When, Whenne, Whereas, Wherefore, Wherfor, Wherfore, Whether, Whil, While, Whiles, Whilom, Why, Whēn, Yea, Yet, Yf

Note: I added a few more Latin conjunctions and transition words after I ran the initial segmentation algorithm. I use all the words above when I further sub-segment the segments, notes, and Bible verses during the text reuse mining procedure.

Algorithm 1 ConstructSegment($S[1..n]$, $doc[1..m]$, t)

Inputs: An ordered list of tokens named S of size n (when combined, it is a "segment"), a list of lists doc of size m such that the last sequence of tokens in doc (i.e., $doc[m]$) is the fully constructed segment immediately preceding S , and a string t representing the next token in the corpus after $S[n]$, which is the last token in S .

Output: List: An updated version of doc if there are no more tokens to process. Else, return a recursive call to process the next token.

Helper Functions and Notes:

- Tokens are words and punctuation within the original text, as well as boundary indicators for a section, page, italicized span (i.e., textual emphasis), note, paragraph, non-Latin gap, or missing/illegible page. Only indicators for italicization, non-Latin gaps, and missing/illegible pages will be retained in the output segment, and a placeholder will take the place of a note in a body segment.
- **CombineSegments**($doc[m], S$) adds the elements of S in order to the end of $doc[m]$, so doc has a resulting size of $m + n$. It returns the combined list.
- **EndsWithNum**(S) returns **true** if the last non-punctuation, non-indicator token of S is a numeral, else **false**.
- **InSameParagraph**($S, doc[m]$) returns **true** if S and $doc[m]$ belong to the same paragraph, else **false**
- **InSameSection**($S, doc[m]$) returns **true** if S and $doc[m]$ belong to the same section, else **false**.
- **IsConjunctionOrTransition**(t) returns **true** if t is a known conjunction or transition word, regardless of case, or if MorphAdorner predicts t to be a conjunction. Else, return **false**.
- **Length**(S) returns the number of non-punctuation, non-placeholder tokens in S .
- **MustContinue**(S, t) returns **true** if $S[n]$ is within a parenthetical, an italicized span with no more than five tokens, or a note. It also returns **true** if the following conditions are both true: t is a numeral and $S[n]$ is not a semi-colon, question mark, forward slash, backward slash, or exclamation point. Otherwise, it returns **false**.
- The EOS (end-of-sentence) marker is a binary label predicted by MorphAdorner.

```
1: if MustContinue( $S, t$ ) = false
   then
2:   next_t ← token after  $t$  (NULL if  $t$  is NULL or there are no more tokens after  $t$ )
3:   if InSameSection( $S, doc[m]$ )=false or InSameParagraph( $S, doc[m]$ )=false or  $S[n]$  indicates a page gap then
4:     doc ← AddSegment( $S, doc$ )
5:   else if  $S[n] \in \{., : ; ? ! / \}$  or the EOS marker of  $S[n]$  is 1 then
6:      $x \leftarrow \text{Length}(S)$ 
7:      $y \leftarrow \text{Length}(doc[m])$  {Note that  $doc[m][y]$  refers to the last token of  $doc[m]$ }
8:     if EndsWithNum( $doc[m]$ ) = true and IsConjunctionOrTransition( $S[1]$ ) = false then
9:        $doc[m] \leftarrow \text{CombineSegments}(doc[m], S)$ 
10:    else if  $S[1]$  is in lowercase or  $doc[m][y] \in \{., : ; / \}$  and  $x \leq 15$  and  $y \leq 15$  then
11:       $doc[m] \leftarrow \text{CombineSegments}(doc[m], S)$ 
12:    else
13:      doc ← AddSegment( $S, doc$ )
14:    end if
15:  else if  $x \geq 15$  and  $S[n]$  is a comma and IsConjunctionOrTransition( $t$ ) = true and next_t is not a numeral then
16:    doc ← AddSegment( $S, doc$ )
17:  else if  $t$  is NULL then
18:    doc ← AddSegment( $S, doc$ )
19:  end if
20: end if
21: if  $t$  is not NULL then
22:    $t \leftarrow$  the token after  $t$ 
23:   return ConstructSegment( $S, doc, t$ )
24: else
25:   return doc
26: end if
```

Algorithm 2 AddSegment($S[1..n]$, $doc[1..m]$)

Inputs: Ibid. for S and doc as ConstructSegment

Output: Ibid. as ConstructSegment

Helper Functions and Notes: Ibid.

```
1:  $x \leftarrow \text{Length}(S)$ 
2:  $y \leftarrow \text{Length}(doc[m])$ 
3: if InSameSection( $S, doc[m]$ ) = true and ( $x \leq 5$  or ( $x \leq 15$  and  $y \leq 15$ )) then
4:    $doc[m] \leftarrow \text{CombineSegments}(doc[m], S)$ 
5: else
6:    $doc.append(S)$ 
7: end if
7: return doc
```

After preprocessing these section units with MorphAdorner and this segmentation pipeline, my resulting corpus consists of 681,210 word types distributed in 154.5 million tokens across 6.8 million body segments, and

223,316 types distributed in 3.9 million tokens across 644,627 marginal notes.²⁹ The average length of a Bible verse is 25.31 tokens, with a standard deviation of 10.68.³⁰ My segmentation algorithm is mostly effective: the average segment length in my corpus is 22.5 tokens, with a standard deviation of 11.3. For the unsegmented marginalia, the average note length is 6.07 tokens, with a standard deviation of 12.16 tokens. Because most marginal notes consist of short citations, I remove all numerals and filter out the overly short segments during quotation mining; for notes that are long passages of text rather than citations, I segment them at a later stage. The body segments and notes I describe in this section are the ones that are displayed on EEPS’ website as individual units, the basis on which I index scriptural citations and quotations. In other words, the reference index I build will display unique segment identifiers rather than page or paragraph numbers. Such a design helps me more precisely localize and analyze the relationship between references and their immediate contexts.³¹

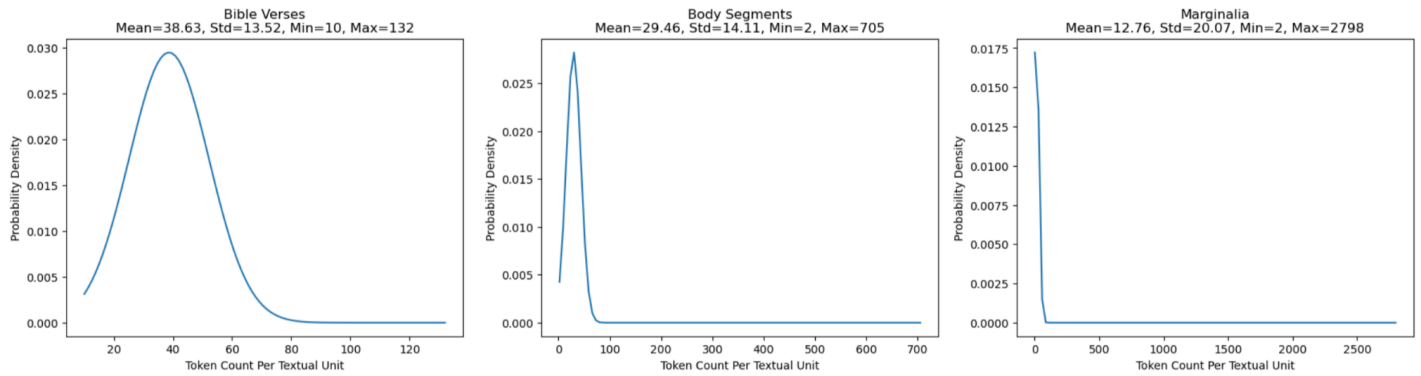


Figure 2: Distributions of the Number of Tokens in Bible Verses, Body Segments, and Marginalia Produced By MacBERTh’s Tokenizer

With MacBERTh’s subword tokenizer, the mean input lengths still remain comparable for segments and verses, while the marginal note lengths also remain less than ideal (see Figure 2). The outliers of overly long segments in my corpus, despite all the edge cases considered by my segmentation algorithm, present a pressing problem because some of them extend beyond the truncation threshold of the tokenizer. The longest outlier is a fifteenth-century passage which has the following structure of conjunctions, transition words, and a single punctuation mark located at the end, totaling 705 subword tokens: “... & ... Thā ... Than ... & ... but ... & ... & ... & ... Thā ... & ... and ... But ... & ... Than ... & ... and ... & ... & ... & ... Than ... & ... & ... & ... & ... & ... But ... & ... & ... & ... And ... & ... & ... & ...”³² Fortunately, segments that are longer than 132 tokens only make up 0.09 percent of all body segments and 0.03 percent of all marginalia. I subdivide these further when examining them for quotations; EEPS’ scriptural index shows the likely Bible verse matches, if any, for the sub-segments of each segment, as well as any associated marginal notes and sub-segments of notes, on its dedicated webpage.

As is clear from Figure 3, a considerable proportion of EEPS’ segments end with commas, semi-colons, colons, and forward slashes, which indicates that these marks frequently appear in overly long sentences, as they do not meet the criteria I designed to merge such segments with

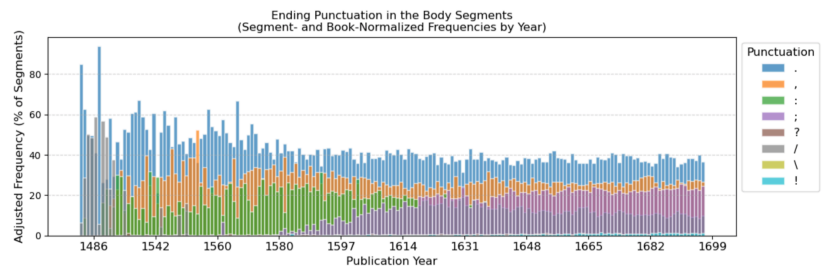


Figure 3

²⁹ All token and type counts exclude punctuation marks, boundary indicators, and placeholders for page gaps, notes, and non-Latin gaps.

³⁰ I already segmented the overly long verses of Ecclesiasticus 1.1 (AKJV), Manasseh 1.1 (AKJV), and Ecclesiasticus 1.1 (Vulgate).

³¹ Segment identifiers take the form of a tuple: a unique EEBO-TCP-assigned ID with six characters, the index of that segment within the book, and its location on a page (either “In-Text” or a “Note #” where “#” indicates the index of that note within the segment).

³² “Thā” and “Than” all mean “then” in this passage. John Mirk, [*Liber Festivalis*], 1486, <http://name.umdl.umich.edu/A07572.0001.001>. Also see <https://www.earlyenglishprintedsermons.org/tcpIDpub/A07572/references>.

their immediately preceding neighbors.³³ In seventeenth-century sermons, the usage of the semi-colon actually overtakes that of the colon. Forward slashes as segment dividers occur exclusively in the few sermons from the fifteenth- and early sixteenth-century, which indicates that only these earliest books contain adaptations of sermons into verse. Unsurprisingly, backward slashes never appear. Both the exclamation point and question mark are relatively rare segment delimiters. Figure 4 reveals that less than 7 percent of all question marks conclude segments; since they often end clauses that are brief and followed by lowercase words, they would be merged with a prior segment, likewise short. In these sermons, interrogative clauses function much more often as subordinates to their immediate textual neighbors rather than standalone sentences. Because nearly 60 percent of exclamation marks function as segment delimiters, their rarity in Figure 3 indicates that they are generally uncommon in the corpus. Most commas do not end segments, which provides assurance that the segmentation algorithm is not splitting up conjunctive phrases. Many periods appear in abbreviations and after numerals, so it is reasonable that more than half of them remain in the middle of the resulting segments; the situation is similar for slashes.

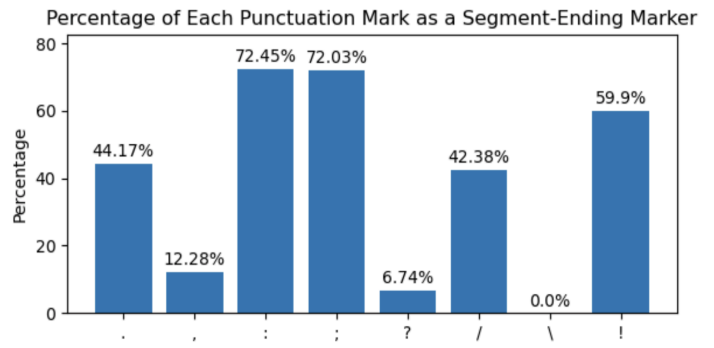


Figure 4

Conjunctions are by far the most commonly occurring segment-initial words. These sermons' paratactic nature is evident through the relatively stable green and purple bars in Figure 5 that respectively represent the upper- and lower-case "and," the latter which I allow as an initial marker only if it follows a comma in a long sequence and is not itself followed by a numeral (thus ensuring that it does not belong to some citation). On the other hand, the decline of the capitalized "The" as an initial marker in the marginalia strongly suggests that marginal notes began to primarily perform a citational rather than explicatory role beginning in the late sixteenth century.

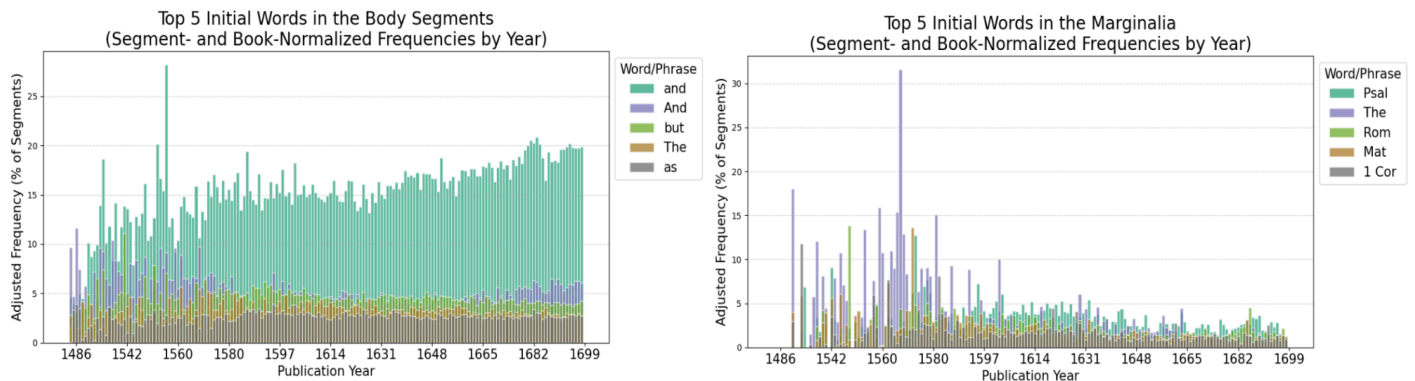


Figure 5

The final part of this corpus overview highlights the special types of text which I expect to be closely connected or overlapping with references: there are 100,866 non-Latin-alphabet gaps, 257,293 foreign-language spans, and over 6 million italicized spans in the body segments and marginalia combined, excluding any "trivial" spans that have no non-placeholder, non-indicator, non-numeral, and non-punctuation tokens. The sheer enormity of the count of italicized spans is due to the fact that sites of textual emphasis can range from a few non-trivial tokens to a long passage, the latter type which I divide across consecutive segments; in other words, many discrete italicized spans in my corpus are actually continuous in their original transcriptions. Although there is a small but steady increase in expressions written in non-Latin alphabets (predominantly Greek and Hebrew as expected with the spread of Humanism), we find

³³ All percentages over time in the visualizations of this paper are normalized first by the number of segments in each book and then by the number of books printed in each year to remove biases arising from varying book length and number of books printed per year.

that the Englishness of sermons greatly increases as the eighteenth century approaches—the noticeable decline of foreign spans, predominantly sites of Latinity, is another piece of evidence for Hunt’s aforementioned observation that Latin quotations are more commonly found in earlier Catholic sermons.³⁴ However, there is a noticeable uptick of foreign-language usage in the early seventeenth century, perhaps coincidental with what Hunt characterizes as “a nostalgic folk-memory of pre-Reformation sermons.”³⁵

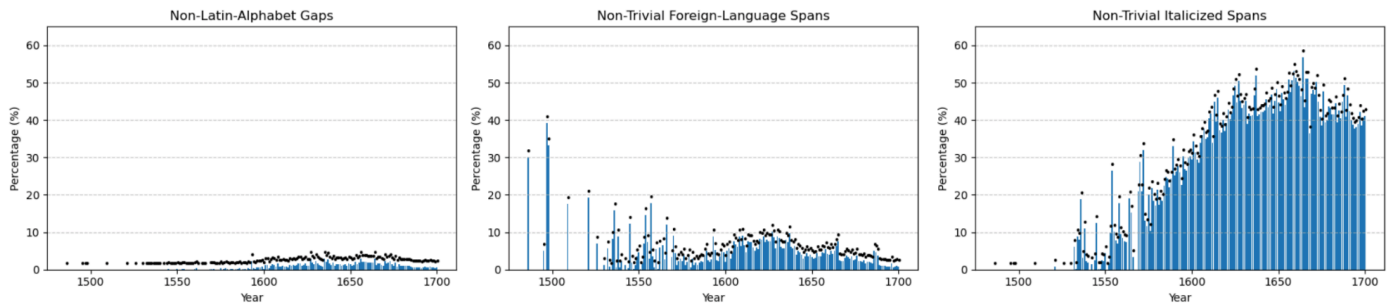


Figure 6: Normalized Percentages of Special Text Spans Per Publication Year (All Segments and Notes)

Eventually, it is for EEPS to verify if quotations from the Latin Vulgate are indeed reappearing before their corresponding English translations in the early seventeenth century. Moreover, Hunt notes that audiences’ preference for sermons on the Gospels rather than the Old Testament is another form of resistance to the Reformation.³⁶ Where and when is such a preference reflected in the individual verses that preachers choose to quote and cite before their audiences?

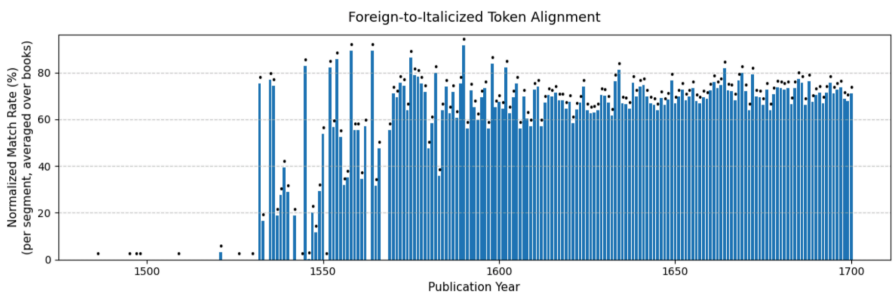


Figure 7: Normalized Percentage Per Publication Year of Foreign Tokens Italicized in Segments w/ Foreign Text

Are there extensive typological references to the Old Testament within sermons on the Gospels? I also calculate the proportion of these special spans which co-occur or are themselves Biblical quotations, which reveals the extent to which it was standard practice for the typesetter to differentiate quotations, especially Latin ones, from other categories of text on the page—an early modern equivalent to our use of quotation marks. On average, 61.7 percent of the foreign-language tokens in segments containing foreign text are italicized, and trends in this practice remain consistent in the seventeenth century, whereas sixteenth-century sermons exhibit polarized preferences for textual emphasis. If we know which particular Bible verses are overwhelmingly unemphasized in these sermons, we would be able to gain insight about what stationers and preachers may have thought readers should accept as everyday language rather than notice as an important proof text.

Related Work

My methodology for text reuse detection is entirely indebted to the rapid advancements in NLP over the past six years, particularly the advent of a family of LLMs derived from BERT, originally authored by Devlin et al. (2019), to produce dense, contextual word embeddings suitable for many classification tasks, such as semantic similarity and

³⁴ See Hunt, *Art of Hearing*, 264-287. Note that a dot in the graphs of Figure 5 represents that there are body segments in a year, even if the percentage is zero. There are no non-Latin-alphabet gaps prior to the late 1540s.

³⁵ Hunt, *Art of Hearing*, 287. He writes that an “unexpected form of lay resistance” to new Protestant styles of preaching “was the demand for learned sermons with a parade of Latin quotations.”

³⁶ Hunt, *Art of Hearing*, 288: “Another form of lay resistance – again, easily overlooked – was a dislike of sermons on the Old Testament, and a demand for sermons on the Gospels.”

named entity recognition.³⁷ Especially important is the subsequent release of the aforementioned SBERT by Reimers and Gurevych in the same year, which brought to the field a suite of efficient implementations, training paradigms, and pre-trained models for sentence-level vectorization and classification, enabling applications to semantic search (i.e., IR) and paraphrase detection.³⁸ BERT's architecture takes into account a word's left and right context in a sentence; when averaged within SBERT's framework, these vectors are semantically rich sentence-level representations. Since they use subword tokenizers, they are also robust enough to handle texts with minimal preprocessing and normalization. MacLaughlin et al. (2021) compare several pre-trained and fine-tuned SBERT models with other neural networks and lexical approaches on various datasets containing local text reuse, e.g., reused text embedded within a passage that is otherwise unrelated to the source, ultimately recommending researchers to fine-tune their own BERT-based models after filtering a corpus with an approach based on Term-Frequency, Inverse Document Frequency (TF-IDF), which ensures that a word which appears in more documents of a corpus are weighted to be less significant than those that appear exclusively in a few documents.³⁹ After evaluating how thirty-six SBERT models perform on distinguishing how scriptural passages are recontextualized in social media posts, Periti et al. (2024) conclude that off-the-shelf, pre-trained Sentence Transformer (Bi-Encoder) models are stronger overall than their Cross Encoder models.⁴⁰ The Sentence Transformer model produces separate embeddings for each query and passage to be retrieved, whereas the Cross Encoder outputs a single similarity score for each pair of input passages; these two respectively function as the retriever and reranker in a modern IR system. Recent work by Kanerva et al. (2025) demonstrates the feasibility of fine-tuning SBERT models for recognizing spans, rather than entire sentences, of paraphrases embedded within longer passages.⁴¹

Prior to these developments, computational research on the extensive spectrum of intertextuality relied on sequence alignment or similarity-measuring approaches, often referred to as “fingerprinting,” which use lexical or syntactic features such as n-grams or sparse word embeddings measured with TF-IDF. N-grams, also known as “shingles,” can be either contiguous or “skip-grams” depending on whether they use a consecutive or non-consecutive sequence of *n* characters or words. For investigating text reuse in ancient corpora, the seminal quantitative work is the approach by John Lee (2007) which uses TF-IDF vectorization combined with source verse order, proximity, and alternation patterns to explore text reuse within the Gospels of the Greek New Testament.⁴² IR applications in historical text reuse research also appeared early: Bamman and Crane (2008) use lexical similarity, distributional semantics, word order, dependency trees, and metrical patterns to retrieve allusions in Classical Latin poetry.⁴³ Later research by Moritz et al. (2016) also focuses on Bible reuse, but they assess the limitations of automated approaches for capturing the transformation of linguistic features (synonymization, capitalization, and part-of-speech) in scriptural references within Ancient Greek and Latin texts—respectively the works of Clement of Alexandria and Bernard of

³⁷ Jacob Devlin et al., “BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding” (arXiv, May 24, 2019), <https://doi.org/10.48550/arXiv.1810.04805>.

³⁸ Reimers and Gurevych, “Sentence-BERT.”

³⁹ Ansel MacLaughlin, Shaobin Xu, and David A. Smith, “Recovering Lexically and Semantically Reused Texts,” in *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, ed. Lun-Wei Ku, Vivi Nastase, and Ivan Vulić (*SEM 2021, Online: Association for Computational Linguistics, 2021), 52–66, <https://doi.org/10.18653/v1/2021.starsem-1.5>.

⁴⁰ Francesco Periti et al., “TRoTR: A Framework for Evaluating the Re-Contextualization of Text Reuse,” in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, ed. Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (EMNLP 2024, Miami, Florida, USA: Association for Computational Linguistics, 2024), 13972–90, <https://doi.org/10.18653/v1/2024.emnlp-main.774>.

⁴¹ Jenna Kanerva et al., “Semantic Search as Extractive Paraphrase Span Detection,” *Language Resources and Evaluation* 59, no. 1 (March 1, 2025): 257–76, <https://doi.org/10.1007/s10579-023-09715-7>.

⁴² John Lee, “A Computational Model of Text Reuse in Ancient Literary Texts,” in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, ed. Annie Zaenen and Antal van den Bosch (ACL 2007, Prague, Czech Republic: Association for Computational Linguistics, 2007), 472–79, <https://aclanthology.org/P07-1060/>.

⁴³ David Bamman and G. Crane, “The Logic and Discovery of Textual Allusion,” 2008.

Clairvaux manually annotated by Laurence Mellerin (2014) in his Biblindex project.⁴⁴ When Moritz and Steding (2018) again investigate the representativeness of different textual features by applying machine learning classifiers trained on modern English corpora to Biblindex's annotated dataset of scriptural paraphrases in Bernard's works, they conclude that static, dense word embeddings are significantly stronger features for calculating similarity than lexical features for paraphrastic text reuse.⁴⁵ Using the same Bernard corpus, Manjavacas et al. (2019) find that IR models that use TF-IDF for allusion mining actually outperform those that are based on custom scoring functions, static word embeddings, and sentence embeddings calculated by averaging over those word embeddings.⁴⁶ In a rare computational article that focuses primarily on Early Modern literature, Peverelli et al. (2022) cites Manjavacas et al. (2019) as justification for their decision to compute the cosine similarity between each pair of texts within their corpus of transnational Neo-Latin plays, compared with works of Classical Latin drama, by vectorizing each text with TF-IDF.⁴⁷

Sequence alignment algorithms are also popular within publicly accessible tools for humanities data analysis, including: *TRACER*, a suite of implemented algorithms for text reuse based on n-gram features, which Büchler et al. (2014) use for comparing intertexts across seven English Bibles; the pre-2019 legacy documentation of the Classical Language Toolkit for ancient languages which separately demonstrates text reuse mining with edit distance, longest common substring, and minhashing; David Smith's Passim library which looks for overlapping character n-grams; the TextPAIR package of the University of Chicago's ARTFL Project which determines textual similarity by comparing trigrams; Yale DHLab's interactive *Intertext* tool which uses existing implementations of the minhashing algorithm.⁴⁸ Passim's documentation contains a tutorial on using the tool for Biblical text reuse with EEBO-TCP texts, and I accordingly apply Passim with its default parameters to my custom training dataset as a baseline method.⁴⁹ All of these evaluate overlapping sequences of characters or tokens rather than disjoint segments. Smith et al. (2013) look for reprinted or plagiarized articles in nineteenth-century American newspapers digitized by the Library of Congress by evaluating n-gram overlap, research which continued into the next year with an additional analysis of Congressional

⁴⁴ Maria Moritz et al., "Non-Literal Text Reuse in Historical Texts: An Approach to Identify Reuse Transformations and Its Application to Bible Reuse," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, ed. Jian Su, Kevin Duh, and Xavier Carreras (EMNLP 2016, Austin, Texas: Association for Computational Linguistics, 2016), 1849–59, <https://doi.org/10.18653/v1/D16-1190>;

Laurence Mellerin, "New Ways of Searching with Biblindex, the Online Index of Biblical Quotations in Early Christian Literature," in *Digital Humanities in Biblical, Early Jewish and Early Christian Studies*, ed. Claire Clivaz, Andrew Gregory, and David Hamidović (BRILL, 2014), 177–90, https://doi.org/10.1163/9789004264434_012.

⁴⁵ Maria Moritz and David Steding, "Lexical and Semantic Features for Cross-Lingual Text Reuse Classification: An Experiment in English and Latin Paraphrases," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, ed. Nicoletta Calzolari et al. (LREC 2018, Miyazaki, Japan: European Language Resources Association (ELRA), 2018), <https://aclanthology.org/L18-1311/>.

⁴⁶ Enrique Manjavacas, Brian Long, and Mike Kestemont, "On the Feasibility of Automated Detection of Allusive Text Reuse," in *Proceedings of the 3rd Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, ed. Beatrice Alex et al. (LaTeCH 2019, Minneapolis, USA: Association for Computational Linguistics, 2019), 104–14, <https://doi.org/10.18653/v1/W19-2514>.

⁴⁷ Andrea Peverelli, Marieke van Erp, and Jan Bloemendal, "Tracking Textual Similarities in Neo-Latin Drama Networks," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, ed. Nicoletta Calzolari et al. (LREC 2022, Marseille, France: European Language Resources Association, 2022), 5295–5303, <https://aclanthology.org/2022.lrec-1.567/>.

⁴⁸ Marco Büchler et al., "Towards a Historical Text Re-Use Detection," in *Text Mining*, ed. Chris Biemann and Alexander Mehler, Theory and Applications of Natural Language Processing (Cham: Springer International Publishing, 2014), 221–38, https://doi.org/10.1007/978-3-319-12655-5_11; "Multilingual — Classical Language Toolkit Documentation," accessed June 16, 2025, <https://legacy.cltk.org/en/latest/multilingual.html#text-reuse>; David Smith, "Dasmig/Passim," Python, June 12, 2025, <https://github.com/dasmig/passim>; "ARTFL-Project/Text-Pair," Python (2016; repr., ARTFL-Project, April 8, 2025), <https://github.com/ARTFL-Project/text-pair>; "YaleDHLab/Intertext," Python (2017; repr., Yale Digital Humanities Lab, January 24, 2025), <https://github.com/YaleDHLab/intertext>.

⁴⁹ Matteo Romanello and Simon Hengchen, "Detecting Text Reuse with Passim," *Programming Historian*, May 16, 2021, <https://programminghistorian.org/en/lessons/detecting-text-reuse-with-passim>.

statements (Xu et al., 2014).⁵⁰ Using this same source of historical newspaper data, Lincoln Mullen’s *America’s Public Bible* project (2023), which began in 2016, identifies and analyzes scriptural quotations in nearly fifteen million nineteenth- and twentieth-century newspaper pages.⁵¹ Much inspiration for EEPS came from the visualizations of quotation trends over time, the decision to examine different Bible versions, the interactive interface, and the case studies of verse histories in Mullen’s work. Rather than align sequences of lexical features, he relies solely on machine learning classifiers for supervised paraphrase classification, finding that a logistic model, trained with token counts weighted by TF-IDF and the proportion of a matching Bible verse, outperforms neural networks trained using those same features.

Although my method for quotation identification differs greatly from Mullen’s, I adhere to the interpretive approach which he names “disciplined serendipity”: the creation of an interface that allows readers to move effortlessly from a prediction or statistic for a text to its location, situated in its original context. I provide an interactive reference index and custom semantic search engine for scriptural quotations on a dedicated website incorporating EEPS’ mined references and fine-tuned models. Current databases for Early Modern literature, such as EEBO’s interface on ProQuest and the website of the *EarlyPrint* Lab of Northwestern University and Washington University of St. Louis, mainly support lexical search with a known list of spelling variations.⁵² *EarlyPrint* introduced me to MorphAdorner, and I greatly appreciate its interactive visualizations and searchable corpus of annotated texts. In the same spirit, I also publicize my running record of references on this website to offer an interactive research tool of indexed and searchable references, texts, visualizations, and metadata.⁵³ However, *EarlyPrint*’s lack of support for semantic search limits their Phrase Search tool’s usefulness for exploring text reuse; the tool relies on BlackLab’s corpus query language, built fundamentally on regular expressions, token matching, and dependency parsing.⁵⁴ To find semantically similar but lexically different phrases, vectorized search is often necessary, albeit with much greater resource and memory demands. Thus, EEPS also involves the development of a search engine for querying sermon titles and marginalia using suitable SBERT models.⁵⁵

Methodology

For identifying and classifying different types of Bible reuse in these sermons, EEPS relies on a combination of three different model types: a fine-tuned SBERT bi-encoder that uses MacBERTh as its base word embedding model, a fine-tuned SBERT cross encoder likewise built on MacBERTh, and a traditional BM25 approach that uses TF-IDF and document length for retrieval.⁵⁶ Bi-encoders produce separate embeddings for queries and passages to retrieve, whereas the cross encoder architecture produces a single similarity score for a pair of input sentences.⁵⁷ The essence of

⁵⁰ David Smith, Ryan Cordell, and Elizabeth Maddock Dillon, “Infectious Texts: Modeling Text Reuse in Nineteenth-Century Newspapers,” in *2013 IEEE International Conference on Big Data*, 2013, 86–94, <https://doi.org/10.1109/BigData.2013.6691675>; Shaobin Xu et al., “Detecting and Evaluating Local Text Reuse in Social Networks,” in *Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media*, ed. Alice Oh et al. (Baltimore, Maryland: Association for Computational Linguistics, 2014), 50–57, <https://doi.org/10.3115/v1/W14-2707>.

⁵¹ Lincoln A. Mullen, “America’s Public Bible: A Commentary” (Stanford University Press, 2022), <https://americaspublicbibles.org>. See “Methods: The how and the why of finding biblical quotations” (<https://americaspublicbible.supdigital.org/essay/methods/>).

⁵² EarlyPrint Lab, “EarlyPrint,” EarlyPrint, accessed June 16, 2025, <https://earlyprint.org/>; “BCQL | / BlackLab /,” accessed June 16, 2025, <https://blacklab.ivdnt.org/guide/query-language/>.

⁵³ I built the site using Flask as my framework and PostgreSQL for my relational database. I currently host this site using an Ubuntu server from DigitalOcean with 8 CPU cores, 16GB RAM, and 50GB disk storage.

⁵⁴ EarlyPrint Lab, “Phrase Search,” EarlyPrint, January 17, 2014, https://earlyprint.org/lab/tool_phrase_search.html.

⁵⁵ I do not store the embeddings for body segments due to resource constraints.

⁵⁶ For BM25, see its original paper: S. E. Robertson and K. Sparck Jones, “Relevance Weighting of Search Terms,” *Journal of the American Society for Information Science* 27, no. 3 (1976): 129–46, <https://doi.org/10.1002/asi.4630270302>. I use the following Python implementation of Okapi BM25: Dorian Brown, “Dorianbrown/Rank_bm25,” Python, June 20, 2025, https://github.com/dorianbrown/rank_bm25.

⁵⁷ “Semantic Search — Sentence Transformers Documentation,” accessed June 10, 2025, https://www.sbert.net/examples/sentence_transformer/applications/semantic-search/README.html.

fine-tuning is contrastive learning, a strategy of minimizing the error (“loss”) of a model when distinguishing between true positives and different types of negatives for each input query.⁵⁸ Each training step updates the model’s weights so that it can produce sentence embeddings that are closer in vector space for positives rather than negatives. The incorrect samples in this case can either be in-batch negatives, which are the positives for different query passages in the same batch of a predefined size, or hard negatives, which are the top-ranking irrelevant passages returned by the off-the-shelf version of MacBERTh. I use batches of size 64 with 1 hard negative per query, so the model encounters 1 positive and 64 negatives per query. I do not use more hard negatives because these models are particularly prone to overfitting on my training data; a second training round (“epoch”) with a different set of hard negatives lowers the model’s performance. As such, I ensure that no parallel texts or cross references of the same verse are in the same batch using a conflict graph built using a greedy coloring algorithm.⁵⁹ By default, I withhold a random 10 percent of the training corpus to use as a development dataset, which is separate from the test set that I use to evaluate the model’s generalizability. Any batches of the training set that cannot reach a size of 64 without encountering conflicts, e.g., two parallel verses become in-batch negatives for each other, are likewise added to the development set.

To reduce ambiguity and increase the models’ sensitivity to scriptural citations in the input passages, I prepend the book, chapter, and verse number to each individual Bible verse, which forms the corpus of passages for a bi-encoder to retrieve; I remove several hundred verses that are overly vague, particularly verses that only contain formulaic language introducing some speaker, e.g., “Ezekiel 28.1: The word of the Lord came againe vnto me, saying.” Because MacBERTh uses a subword tokenizer and one of my research objectives is to identify the uses of different Bible versions, I do not use the regularized spellings or lemmata from MorphAdorner for either the input passages or Bible verses.⁶⁰ Instead, I strip placeholders and indicator tokens and convert text to lower case.

I train the bi-encoder and cross-encoder with the same training set. My training corpus, which has nearly sixty thousand unique samples, combines several different sources of texts and noise. I artificially create noisiness in order to mimic true cases of scriptural reuse, which may be surrounded by citations and spans of non-reuse. For example, the citations in a given segment may actually be associated with a prior or following segment instead, so an ideal model should learn to pay attention to them but not be biased to predicting a high similarity score between a scriptural citation and the prepended verse numbers in the Bible corpus. Justification for augmenting queries with noisy text comes from the work on paraphrase span detection by Kanerva et al (2025).⁶¹ Because the following sets vary greatly in size, I choose to train my models with them all at once rather than in separate rounds to prevent them from forgetting what they learned in prior rounds or

1. **Parallel verses:** The different Bibles are natural sources of text reuse data. I identify parallel verses from the six different Bibles using STEP Bible’s compact tests for mapping different versification traditions, the most

⁵⁸ EEPS uses the MultipleNegativesRankingLoss for the bi-encoder and the BinaryCrossEntropyLoss for the cross encoder. See “Loss Overview — Sentence Transformers Documentation,” accessed June 20, 2025, https://sbert.net/docs/sentence_transformer/loss_overview.html for loss functions of the bi-encoder. See “Loss Overview — Sentence Transformers Documentation,” accessed June 20, 2025, https://sbert.net/docs/cross_encoder/loss_overview.html for loss functions of the cross encoder.

⁵⁹ For graph coloring, see “Graph Algorithms,” <https://www.cs.cornell.edu/courses/cs3110/2012sp/recitations/rec21-graphs/rec21.html>.

⁶⁰ This is not to say that adornment is not useful for anything in this project besides segmentation. Lemmata, regularized spellings, and part-of-speech information are valuable features for sequence labeling tasks such as named entity recognition and citation span detection, which the EEPS project will eventually tackle.

⁶¹ Kanerva et al., “Semantic Search as Extractive Paraphrase Span Detection.”

common here being the English KJV, Hebrew, and Latin traditions.⁶² An interesting result here is that the Geneva Version has the same versification pattern as the French NEG 1979 version for Job 40 rather than the English, Hebrew, Latin, or Greek versions. These parallel texts form the basis of three different types of queries: (1) **full** parallel verses (hereafter the acronym **FPV**), (2) **partial** parallel verses segmented by punctuation (**PPV**), and (3) **hybrid** combinations of English and Latin verse text (**HYB**). I let the “master id” for each group of parallel texts of a verse be its AKJV verse id. In total, there are 30,981 AKJV verses with parallel text mappings.

2. **Biblical Proper Nouns** (hereafter **PN**): To train the model to recognize allusions, I compile sequences of proper names and capitalized words in each parallel verse of the references with proper names indexed by STEP Bible only if that verse has a named entity that occurs in no more than ten individual verses, since ten is the mean number of references associated with each entity.⁶³ STEP Bible’s dataset is for the English Standard Version of 2001, the KJV, and the New International Version, and I assume that the verse numbers align with the AKJV. To align these entities with the other Bibles, it is necessary to find the corresponding noun phrases in parallel Bible texts. For multi-word expressions and lower-case words, I use Python’s `diffib` `SequenceMatcher` function with a similarity threshold of 0.7 to examine a range of n-grams from single words to 5-grams. I also use the off-the-shelf `MacBERTh` to find semantic matches within the parallel verses of each reference of the two aforementioned types of noun phrases. I only include the first lexical or semantic match of a noun for each reference of that noun. For the remaining references associated with at least one noun phrase, I include all capitalized text in their parallel verses, even if such words may not strictly correspond to entries within this set of proper noun phrases. There are 3,572 master ids with these sequences, comprising 14 thousand verses in the corpus of Bibles.
3. **Cross References (CR)**: There is a comprehensive collection of 340,000 Biblical cross references within the English Standard Version from the OpenBible site.⁶⁴ Again, I make an assumption that the versification generally aligns with the King James Version, and I accordingly map these cross references onto each different Bible version using my dataset of parallel verses. There are 31,108 master ids with known cross references, comprising 131,546 total verses.
4. **Quotations and Paraphrases from sermons (QP)** comprises 3,873 segments and notes with a known scriptural quotation or paraphrase identified using off-the-shelf `MacBERTh` and then manually verified.
5. **Instances of Non-Reuse from sermons (NON-QP)** comprises 24,185 marginal notes with no scriptural text reuse, and most of them contain scriptural citations. I use a placeholder empty string as the positive passage for each of these examples.
6. **Most Frequent Bible Book Variants (BBV)** is a collection mapping each Bible book name to its most commonly used form within scriptural citations in EEPS’ corpus of sermons. Compared to classical, patristic, or contemporary book references, the form and structure of scriptural citations tend to be highly predictable, especially in the seventeenth century; my algorithm for identifying and standardizing these citations accounts for different arrangements of numerals, periods, commas, ampersands, and hyphens. Using edit distance

⁶² I had to manually convert these tests, written in plain text, into a generalizable code program to apply these rules automatically to any input set of Bibles; that was a time-consuming and rather challenging task. See STEP Bible, “STEP Bible-Data/TVTMS - Translators Versification Traditions with Methodology for Standardisation for Eng+Heb+Lat+Grk+Others - STEP Bible.Org CC BY.Txt at Master · STEP Bible/STEP Bible-Data,” GitHub,

<https://github.com/STEPBible/STEPBible-Data/blob/master/TVTMS%20-%20Translators%20Versification%20Traditions%20with%20Methodology%20for%20Standardisation%20for%20Eng%2BHeb%2BLat%2BGrk%2BOthers%20-%20STEPBible.org%20CC%20BY.txt>.

⁶³ STEP Bible, “STEP Bible-Data/TIPNR - Translators Individualised Proper Names with All References - STEP Bible.Org CC BY.Txt at Master · STEP Bible/STEP Bible-Data.”

<https://github.com/STEPBible/STEPBible-Data/blob/master/TIPNR%20-%20Translators%20Individualised%20Proper%20Names%20with%20all%20References%20-%20STEPBible.org%20CC%20BY.txt>

⁶⁴ *Bible Cross References*. <https://www.openbible.info/labs/cross-references/>. See the visualizations on their home page.

metrics, I compiled over a thousand probable Bible book variant spellings and abbreviations, and I only identify a span as a scriptural citation if it contains one of these keywords followed by at least one numeral. Note that false positives may include Biblical commentaries whose titles include Bible chapter or verse numbers, but I do not differentiate these from individual Biblical citations at this stage of EEPS. The recognition and analysis of citations and named entities, especially non-scriptural instances, are important downstream applications of linguistic adornment and Bible reuse detection. For the purposes of this paper, I only explore preachers’ citational habits and errors relative to their explicit reuse of scriptural text.

Of these sets, only the following are queries: FPV, PPV, HYB, PN, QP, and NON-QP. For each general verse identifier from the AKJV, I randomly choose its query form as a FPV, PPV, HYB, or PN from the known parallel verses of that text. Sources of noise are CR, NON-QP, and BBV; for each query, I randomly choose to append no noise, the text of a cross reference, a sequence of citational references to a query’s cross references, or a short NON-QP span. For the resulting query text, I rely on more randomness when choosing whether or not to append a correct citation corresponding to the positive passage; this citation can either be a verse or chapter citation, and it uses the book name found within BBV. Moreover, the citation may be accompanied with some variant of “as in,” “see,” and “vid.” as is found in actual references within these sermons. I allow all known parallel texts of a positive for a query to count as correct answers.

Model	Type	FP@1	R@6	R@25	MAFP@6	%Hit
Okapi BM25	QP	0.522	0.245	0.338	0.417	98.24
	NON-QP	0.006	0.024	0.085	0.011	00.61
paraphrase-multilingual-mpnet-base-v2 (Paraphrase Detection SBERT)	QP	0.379	0.201	0.292	0.313	93.29
	NON-QP	0.081	0.143	0.216	0.102	08.14
multi-qa-mpnet-base-cos-v1 (Question-Answering SBERT)	QP	0.439	0.241	0.355	0.377	92.12
	NON-QP	0.087	0.139	0.184	0.105	08.71
msmarco-distilbert-cos-v5 (MSMARCO - IR SBERT)	QP	0.361	0.170	0.240	0.287	94.00
	NON-QP	0.057	0.095	0.144	0.070	05.65
all-mpnet-base-v2 (General Purpose SBERT)	QP	0.401	0.215	0.322	0.339	88.35
	NON-QP	0.113	0.143	0.185	0.123	11.28
MacBERTh (Off-the-Shelf)	QP	0.296	0.164	0.232	0.244	88.24
	NON-QP	0.866	0.910	0.949	0.882	86.62
EEPS_ALL_NO_NOISE_MacBERTh_Epoch1	QP	0.768	0.589	0.806	0.722	98.24
	NON-QP	0.963	0.982	0.990	0.969	96.29
EEPS_ALL_NO_NOISE_MacBERTh_Epoch2	QP	0.768	0.591	0.804	0.722	98.35
	NON-QP	0.572	0.980	0.991	0.809	57.24
EEPS_ALL_QP_MacBERTh_Epoch1	QP	0.815	0.619	0.834	0.756	98.82
	NON-QP	0.894	0.950	0.979	0.914	89.41
EEPS_ALL_MacBERTh_Epoch1	QP	0.820	0.642	0.845	0.766	98.59
	NON-QP	0.964	0.980	0.989	0.970	96.36

Table 1: Bible Verse Retrieval Evaluation

My test set consists of 850 QP and 2,615 NON-QP that do not overlap with those in the training set, and they were identified using MacBERTh as well as fine-tuned bi-encoders before being manually verified and labeled. As with the training and development sets, parallel texts of a positive verse are considered correct. For bi-encoder evaluation, I measure Recall @ k, Fuzzy Precision @ k, and Mean Average Fuzzy Precision @ k.⁶⁵ I use the term “fuzzy” because I allow known cross references of the positives for a query to count as approximate positives when measuring precision.

⁶⁵ Recall is the number of correct answers recounted within the top k results; precision is the proportion of the top-k results that are positives. Mean average precision is the mean of the averaged precision of each rank from 1 to k for each query.

This decision to be flexible with parallel texts and cross references is because the Bible is highly intratextual, and the labeled positives for a true instance of scriptural text reuse in these sermons may not be comprehensive.

Applying Passim with its default parameters to the test set and corpus of Bibles only identifies 3.2 percent of the QP in the test set, albeit with perfect precision.⁶⁶ For each of the IR models in Table 1, I set the boundary between a positive and negative prediction to be the mean minus the standard deviation, essentially the lower bound, of all cosine similarity scores between queries and their retrieved passages in the QP set; this threshold makes sense because most of the similarity scores are normally distributed within each subcorpus of the test set. In other words, each of the models above acts as a binary classifier by predicting a query passage to be a non-reuse instance if the similarity score between the query and a retrieved passage is lower than that threshold. The “%Hit” metric measures the proportion of instances when the highest ranking passage of a query is classified correctly relative to the threshold, and this is the main metric by which I show that using a sequence alignment tool like Passim is not ideal for Early Modern English corpora. BM25 successfully recognizes over 98 percent of the QP set, outperforming off-the-shelf MacBERT_h and four of the strongest SBERT pretrained models for different sentence similarity tasks.⁶⁷ However, the algorithm suffers from mediocre fuzzy precision and recall, and it cannot differentiate between QP and NON-QP. However, the NON-QP set is especially challenging for BM25 because many of those examples also contain scriptural citations, which would lexically overlap with the prepended verse ids in the corpus of Bibles. MacBERT_h outperforms the four SBERT models and BM25 in NON-QP detection significantly, but it has difficulty retrieving the correct labels for QP. My fine-tuned models have names that start with the prefix “EEPS,” and Table 1 shows that the optimal settings are training for only one epoch, using all available sources of query data, and adding noise to those queries.⁶⁸

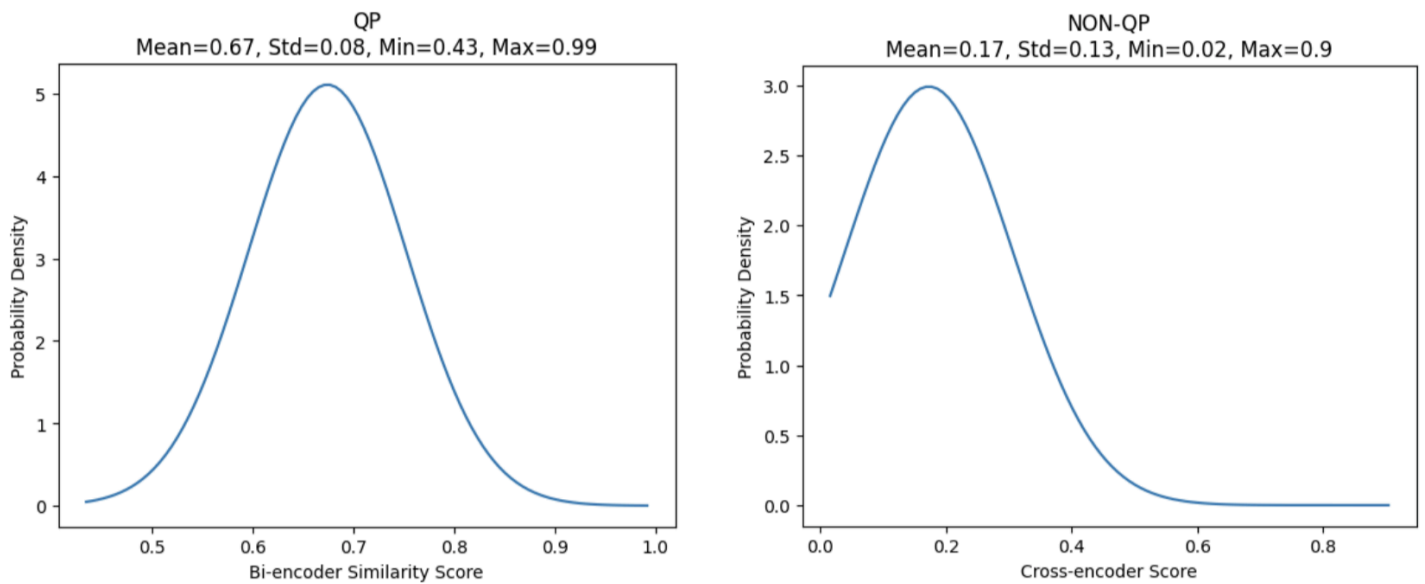


Figure 8: Score Distribution for Passages and Predictions in the QP and NON-QP sets using EEPS_ALL_MacBERT_h_Epoch1 and EEPS_cross-encoder_checkpoint500

Afterwards, I train and evaluate how reranking the top results from my best-performing retriever alters performance. Compared to a bi-encoder, the cross encoder takes significantly more time to run on large datasets because it calculates a score by turning every pair of input passages into a single embedding rather than searching for

⁶⁶ Passim’s EEBO-TCP tutorial for Bible text reuse detection treats each TCP text as an input passage and each Bible verse as an individual document, and its authors do not standardize spellings or do any particular pre-processing. See Matteo Romanello, Simon Hengchen, and David Smith, “Detecting Text Reuse with Passim 2.0.”

⁶⁷ See “Pretrained Models — Sentence Transformers Documentation,” accessed June 22, 2025,

https://www.sbert.net/docs/sentence_transformer/pretrained_models.html.

⁶⁸ “ALL_QP” refers to training only with FPV, PPV, HYB, and QP, excluding PN and NON-QP. “ALL” refers to training with all possible query types with noise randomly added. See ‘notebooks/EEPS_QP_evaluate.ipynb’ in my repository.

the nearest embedding in vector space. Moreover, I design a new threshold: a given passage P counts as a valid QP of a verse V if their cross encoder score is at least equal to the the average of cross scores for the NON-QP set and their cosine similarity score is at least equal to the lower bound (mean minus standard deviation) of similarity scores for the QP set, or if the cross score is at least equal to the lower bound of cross scores for the NON-QP set and its cosine similarity is not lower than the average similarity score for the QP set (see Figure 8). The cross encoder is susceptible to overfitting, thus memorizing training data rather than generalizing well to unseen input. Performance suffers significantly at 1000 training steps, with 64 examples per step, compared to training with only 500 steps using the same training dataset as the bi-encoder but re-formatted accordingly. Despite the significant decrease in fuzzy precision for the QP set, there is a critical advantage to using a fine-tuned cross encoder because it is more sensitive at detecting irrelevance. Considering that the current training data does not contain labels for the literality of Bible text reuse, I prioritize the exclusion of false positives to ensure that my analysis only considers true quotations rather than mistaken predictions of paraphrases or allusions.

Model (Reranking the top 6)	Type	FP@1	R@6	MAFP@6	%Hit
cross-encoder/ms-marco-MiniLM-L6-v2 (SBERT)	QP	0.752	0.642	0.628	87.76
	NON-QP	0.985	0.998	0.989	98.51
EEPS_cross-encoder/checkpoint-500 (fine-tuned MacBERT _h)	QP	0.749	0.642	0.621	84.35
	NON-QP	0.991	0.997	0.992	99.08
EEPS_cross-encoder/checkpoint-1000 (fine-tuned MacBERT _h)	QP	0.700	0.642	0.578	79.05
	NON-QP	0.988	0.995	0.990	98.81

Table 2: Bible Verse Retrieval Evaluation with EEPS_ALL_MacBERT_h_Epoch1 and Different Cross Encoders

During the actual text reuse detection phase, I again divide marginal notes, long body segments, and long Bible verses by examining segment length, punctuation, conjunctions, and transition words. I only consider textual units that have at least three tokens separated by white spaces. The text reuse procedure which I ultimately applied to the corpus is as follows:

1. Semantic search of the **top 6 matches** for a given query passage within the corpus of **full** verses
2. Calculation of cosine similarity scores between the query and the associated **partial units of each full verse** returned by the reranker, as well as any **cited verses in the same or an adjacent textual unit**.
3. Calculation of the **cross encoder score** between the query and each Bible verse, as well as with each of its parts, and any missing cited verses
4. Calculation of the **BM25 scores** for *ibid.* to enable the exploration of literality
5. **Filter results** by the thresholds and include only the **most similar part or whole** of a positive verse

This algorithm identifies **8,091** out of 644,627 marginal notes (**1.25%**) and **1,255,375** out of **6,864,736** body segments (**18.29%**) as vehicles for Bible text reuse in this corpus.

Website Outline

EEPS' website comprises very many components:

- a comprehensive **catalog** of books with sermon-related sections embedded with links to individual references index pages for each available type of text and metadata, as well as to each book's original EEBO-TCP item. This page also contains a semantic search bar for the catalog such that each book is represented by a dense vector of its title and subject headings, as well as relevant documentation. The individual columns of the catalog table also have options for exact lexical search. (<https://www.earlyenglishprintedsermons.org/catalog>)

- A page with visualizations and tables of the available types of **metadata**, complete with a manicule in each row that links to the index of scriptural references for each individual item.
(https://www.earlyenglishprintedsermons.org/metadata_visualization)
- **Corpus statistics** for each book and type of metadata on their respective pages, with embedded links to their reference indices. These pages visualize the distribution of Bible versions among the top QP predictions and also contain tables for applicable *features* for each subcorpus, such as the percentage of units with QP and adjacent matching citations (i.e., quotations accompanied by an exact citation), the percentage of units that do not exhibit any scriptural text reuse, the percentage of units with QP and foreign italicized text, etc.⁶⁹ Ultimately, this forms the basis for any interpretations I might make to answer research questions about Latinity, textual emphasis, and originality.
(https://www.earlyenglishprintedsermons.org/textual_features/Publication/originality)
- The overall **scriptural index** that allows users to query by a particular book, chapter, or verse (e.g., “Genesis 1.1”, “Genesis 1.1 (Vulgate)”, or “Genesis 1 (ODRV)” or “Genesis 1”) to see its adjacent references (citations and QP located within the same or immediately neighboring body segment or its notes), its parsed scriptural citations, and all instances of it as an above-threshold QP prediction regardless of whether it is the highest scoring one for a segment. Only top-scoring QP predictions are included in the adjacent reference index, so if a prediction of “Genesis 1.1 (ODRV)” as “Genesis 1.1 (AKJV)” as an adjacent reference, that means the latter is a higher scoring match than the former for the corresponding text. On the landing page of this index, there is an interactive visualization of the most prominent scriptural books, chapters, and verses over time visualized as pie charts for each part (e.g., “New Testament (ODRV)” for QP and “New Testament” for citations). See <https://www.earlyenglishprintedsermons.org/index>.
- Two **semantic search** engines for the Bibles and for the marginalia. I use my fine-tuned models for the former but a general-purpose SBERT pre-trained model for the latter because my custom models were not trained for

⁶⁹ FEATURE_DESCRIPTIONS = {

```

'cited': "Percentage of units with QP and an adjacent citation",
'cited_exact': "Percentage of units with QP and an adjacent matching citation",
'nonLatin_QP': "Percentage of units with QP and an adjacent NonLatinAlphabet placeholder",
'originality': "Percentage of units that do not exhibit scriptural text reuse",
'Foreign': "Percentage of units with foreign text",
'NonLatinAlphabet': "Percentage of units with a NonLatinAlphabet placeholder",
'Italicization': "Percentage of units with italicized spans of text",
'sim_score': "Average cosine similarity score of top Bible verse predictions per unit",
'cross_score': "Average cross encoder score of top Bible verse predictions per unit",
'near_quotations': "Percentage of units that have high lexical similarity with their Bible verse predictions (any type of score greater than the mean + standard deviation of that score type)",
'foreign_cited': "Percentage of units with QP, foreign text, and an adjacent citation",
'foreign_cited_exact': "Percentage of units with QP, foreign text, and an adjacent matching citation",
'foreign_italicized': "Percentage of units with QP and foreign italicized text",
'foreign_italicized_cited': "Percentage of units with QP, italicized foreign text, and an adjacent citation",
'foreign_italicized_cited_exact': "Percentage of units with QP, italicized foreign text, and an adjacent matching citation",
'foreign_latin': "Percentage of units with Latin Bible QP and foreign text",
'foreign_latin_cited': "Percentage of units with Latin Bible QP, foreign text, and an adjacent citation",
'foreign_latin_cited_exact': "Percentage of units with Latin Bible QP, foreign text, and an adjacent matching citation",
'foreign_latin_italicized': "Percentage of units with Latin Bible QP and italicized foreign text",
'foreign_latin_italicized_cited': "Percentage of units with Latin Bible QP, an adjacent citation, and italicized foreign text",
'foreign_latin_italicized_cited_exact': "Percentage of units with Latin Bible QP, an adjacent matching citation, and italicized foreign text",
}

```

the purpose of asymmetrically searching for general, non-Biblical keywords in text, such as searching for “philosopher” in the marginalia (which returns notes citing Plato and Aristotle). The page itself contains more documentation on the score types, the models, and download options. (<https://www.earlyenglishprintedsermons.org/search>)

Search phrase: I looke through a dark glasse						
Verse & Version	Part Index (blank for full verses)	Text	Cosine Similarity Score	Cross Encoder Score	Okapi BM25 Score	Above threshold
1 Corinthians 13.12 (Geneva)	0-0	1 corinthians 13.12: we see through a glasse darkely:	0.794	0.9	0.89	True
1 Corinthians 13.12 (AKJV)	0	1 corinthians 13.12: for now we see through a glasse, darkely:	0.779	0.866	0.89	True
1 Corinthians 13.12 (Tyndale)	0	1 corinthians 13.12: now we se in a glasse even in a darke speakyng:	0.729	0.621	0.795	True
1 Corinthians 13.12 (ODRV)	0-0	1 corinthians 13.12: by a glasse in a darke fore:	0.716	0.845	0.84	True
1 Corinthians 13.12 (Vulgate)	0	1 corinthians 13.12: videmus nunc per speculum in aenigmatate:	0.703	0.2	0.0	True
Job 29.3 (Douay-Rheims)	0-0	job 29.3: when his lamp shined over my head	0.359	0.033	0.0	False

Showing 1 to 6 of 6 entries

Download as CSV

Previous 1 Next

Search phrase: I looke through a dark glasse		
Text	Cosine Similarity	View Segment and Notes (Indexed by TCP ID and Segment Number)
a looking glasse for beaurtie.	0.593	(A01885,19/5)
when men on earth doe rather behold as in a glasse.	0.565	(A07647,1645)
a darke roome.	0.54	(A12473,35)
a true glasse to see thy sinne in.	0.527	(A13538,6153)
the sight of god.	0.523	(A12995,27)
the eyes of god.	0.519	(A12995,19)
e * > lookeh at things within the *a*le.	0.513	(A13535,603)
how doeth a sleeping man see.	0.504	(A77480,137)
corporal eyes cannot se to much.	0.504	(A05143,7338)

- **Clustering** of books or a type of metadata by a vectorized representation of each item according to the quotational prominence (ORCP for QP) of each book and part from each Bible version, which means a vector of 353 percentages reduced to 3 dimensions for visualization using a standard dimensionality reduction technique. Two dots representing two books are close in this space if they quote prominently from many of the same Bible book of a given version. These visualizations are interactive, and the user can search for the nearest neighbors of some target. The name of each node (e.g., the TCP ID for a book or the full catalog name of an author) is visible upon hovering over that node. <https://www.earlyenglishprintedsermons.org/clusters/Publications>.
- **Scriptural reference indices** with downloadable tables for each publication, each body segment and associated marginalia, and each type of metadata showing statistics (the aforementioned features in addition to reference prominence, diversity, and evenness for each Bible part, book, chapter, and verse), parsed citations, and QP predictions. Links to these pages are embedded in the catalog, the metadata compilation page, and individual publication or textual unit pages.
 - An example for an entire book: <https://www.earlyenglishprintedsermons.org/tcpIDpub/A41135/references>
 - A body segment + associated marginalia with a convenient semantic search bar at the bottom of the page for querying the Bibles: <https://www.earlyenglishprintedsermons.org/segment/A41135/1>
 - An author: <https://www.earlyenglishprintedsermons.org/author/Fenner%2C%20William%2C%201600-1640/references>
- The **full catalog** of body segments and marginalia in two separate tables for the **relevant sections** of a book, as well as a chart of all section divisions with extracted sections in bold. I include information of the page/image, paragraph, and section for each textual unit. As an example, see <https://www.earlyenglishprintedsermons.org/A41135>.

Discussion

Although I have raised many questions about these sermons based on the work of prior scholars, I do not answer any of them in this paper because EEPS’ datasets and methodology are far from maturity after one year’s progress. The approach to scriptural text reuse described here is significantly more sophisticated and reliable than my first attempt in the summer of 2024, when I relied on using SBERT’s pre-trained models to create training data for

MacBERTh and when I did not notice the couple hundred non-relevant publications in my corpus. Since then, I have trained countless models and gradually built my labeled training, development, and evaluation sets, for a time even trying to be independent of SBERT's framework by using the bi-encoder implementation that I wrote for a computer science course assignment. I did not even think of using parallel verses and cross references as training data until March 2025, and I consumed hundreds of compute units on Google Colab training subpar models, vectorizing Bibles and texts, and unwisely doing full mining run-throughs on the entire corpus before fully documenting my methodology, as the writing process identified problems to address and areas for improvement.

EEPS is an ever-evolving and challenging project with tedious and time-consuming demands on me as its principal investigator, machine learning engineer, website developer, and dataset annotator. I am clumsy. For example, despite all my efforts to exclude false positive publications, an advertisement for medical pills written by William Sermon remains in my corpus as the top search result when I query the catalog for "medicine."⁷⁰ A look inside the reference index of that book, which I have retained on the website to use as a cautionary tale, exposes several relatively low-scoring but still above-threshold QP false positives and demonstrates the limitations of using this probabilistic approach for identifying Biblical quotations in non-religious texts.⁷¹ Yet, this example also reminds me that I should be using non-relevant publications as a new source for NON-QP annotations. A key improvement for EEPS in the near future is the curation and annotation of a large, random sample of low-scoring predictions from the current models as exact or near quotations, distant paraphrases, brief allusions, and so on, which enables more fine-grained evaluation for the type and form of text reuse. I am optimistic that I will be ready to explore verse histories and finally articulate literary or historical claims by the next time that I do a full QP mining operation on the entire corpus with models trained and evaluated using a new dataset.

Moreover, for downstream analyses of where and how different Bibles are quoted by preachers, the lack of the full ODRV is a major limitation because the Douay-Rheims version uses modern spelling. This is another reason why the text reuse identification I have done for this paper is far from its final, decisive stage. I intend to soon join in the volunteer efforts to transcribe the entire ODRV; I would experiment with different optical character recognition models to supplement manual transcription. Likewise, it would be better to have the 16th-century original text of the Vulgate rather than the 19th-century edition of it, but I chose the easier option of downloading the text from ebible.org because it is the only Latin Bible version in my dataset.

Further research is required for the likeliest preaching place of a sermon in order to investigate and map the existence of communities with similar sermon and Bible cultures. Eventually, I hope to tackle a fuller scope of references and entities, especially patristic authors, contemporary figures, biblical entities, Hellenistic philosophy, and canon law. To borrow a phrase from Fulton's analysis of John Milton's *Index Politicus* in his book *Historical Milton*, I ultimately envision the EEPS project to be an investigation of whether a "rare window into a writer's private intellectual history" for each of these early preachers can be opened from their extant sermons.⁷² Moreover, Fulton's remarkable quotation from Erasmus' *De Copia* also serves as an intellectual impetus for EEPS' development and future directions: who is quoting what "from the annals of the barbarians, and in fact from the common talk of the crowd"?⁷³ Finally, text reuse detection of other texts, particularly drama, might also uncover extracts with semantic or stylistic similarities—for example, Hunt notes how one printed sermon contains an "immediately recognisable" paraphrase of Shakespeare (172).

⁷⁰ The mention of the author's name in the title is how it turned up in the candidates for this corpus. On another note, I do take some reassurance from the fact that the search for "medicine" does return very interesting sermon titles containing medical and pathological metaphors.

⁷¹ <https://www.earlyenglishprintedsermons.org/tcpIDpub/B05801/references>

⁷² Thomas Fulton, *Historical Milton: Manuscript, Print, and Political Culture in Revolutionary England* (Amherst: University of Massachusetts Press, 2010), 50.

⁷³ Desiderius Erasmus Roterodamus, *De Utraque Verborum ac Rerum Copia*, quoted in Fulton, *Historical Milton*, 56.

However, the most prominently missing component of EEPS' current form is the detailed documentation and improvement of my methodology for identifying and standardizing scriptural citations. My ambition to train conditional random fields for citation span labeling using neural and manually curated features, and then to fine-tune a small generative large language model with a few billion parameters (such as Qwen2.5⁷⁴) to parse and standardize those citations, became too much to handle at the same time as scriptural text reuse detection because there is little overlap between the two.⁷⁵ Moreover, I want to use top QP predictions as a training feature for helping a model learn whether a sequence of tokens comprises some kind of citation, scriptural or otherwise. I soon realized that the wisest decision in the long run is to label and standardize *all named entities* and scriptural, patristic, classical citations in each randomly sampled sermon section in *a single pass* rather than return to these texts in the future out of my aforementioned interest in all explicitly and implicitly referenced sources. The sequence labeling paradigm for Named Entity Recognition and citation identification is the same, and ultimately what I am working on is an Early Modern English equivalent of the CoNLL-2003 dataset.⁷⁶

I am certain that the application of state-of-the-art techniques of supervised machine learning to Renaissance studies is an exciting, albeit challenging, way to uncover new and useful knowledge about large corpora and historical periods. However, it takes very much time, patience, and painstaking detail comparable to the rigor of close reading. Indeed, what EEPS fundamentally needs me to do is more close reading in order to curate more robust and representative for different types of textual features and rhetorical devices, whether scriptural quotations or abbreviated mentions of an ancient author.

Appendix

Like all NLP software, MorphAdorner is not perfect, so I supplement its corpus of standard spellings using WordNet from Princeton University, known Biblical proper nouns from STEP Bible.org, canonical authors from the Cited Loci project, and over sixty thousand nouns and verbs in the corpus checked using OpenAI's GPT-3.5 model.⁷⁷ The EEPS project will primarily use these standardized spellings as features for a later-stage citation identification model, and not at all when identifying citations with heuristics and finding quotations using fine-tuned language models. EEPS' algorithms and models need to capture both morphological and syntactic diversity in a corpus of texts with 172 different publication years, especially since one of my primary objectives is to identify which Bible version is closest in content and form to a given Biblical quotation. Therefore, EEPS primarily focuses upon the original texts with limited preprocessing, such as removing the aforementioned boundary indicators and placeholders and normalizing Unicode characters.

⁷⁴ Qwen Team, "Qwen2.5: A Party of Foundation Models!," Qwen, September 19, 2024, <https://qwenlm.github.io/blog/qwen2.5/>.

⁷⁵ For a good introduction to conditional random fields and their application in the digital humanities, see Matteo Romanello, Federico Boschetti, and Gregory Crane, "Citations in the Digital Library of Classics: Extracting Canonical References by Using Conditional Random Fields," in *Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries (NLP4DL)*, ed. Min-Yen Kan and Simone Teufel (Suntec City, Singapore: Association for Computational Linguistics, 2009), 80–87, <https://aclanthology.org/W09-3610>.

⁷⁶ See "CoNLL 2003 Dataset | Papers With Code," accessed June 30, 2025, <https://paperswithcode.com/dataset/conll-2003>; Erik F. Tjong Kim Sang and Fien De Meulder, "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition" (arXiv, June 12, 2003), <https://doi.org/10.48550/arXiv.cs/0306050>.

⁷⁷ For WordNet, see Princeton University, "About WordNet," Princeton University, 2010, [https://wordnet.princeton.edu/homepage](https://wordnet.princeton.edu/homepage;); "NLTK :: Sample Usage for Wordnet," accessed June 8, 2025, <https://www.nltk.org/howto/wordnet.html>. For Biblical proper nouns, see "STEP Bible/STEP Bible-Data" (2018; repr., STEP Bible, May 25, 2025), <https://github.com/STEPBible/STEPBible-Data>; I used the "[TIPNR – Translators Individualised Proper Names with all References - STEP Bible.org CC BY.txt](#)" file. For the Cited Loci project, see Matteo Romanello, "Cited Loci," Cited Loci, accessed June 8, 2025, <https://citedloci.org/>; I use the following authors dataset from the Cited Loci project: "CitationExtractor/Citation_extractor/Data/Authors.Csv at Master · Mromanello/CitationExtractor," GitHub, accessed February 17, 2024, https://github.com/mromanello/CitationExtractor/blob/master/citation_extractor/data/authors.csv; OpenAI, "GPT-3.5 Turbo Fine-Tuning and API Updates | OpenAI," accessed June 8, 2025, <https://openai.com/index/gpt-3-5-turbo-fine-tuning-and-api-updates/>. See the '[lib/spelling.py](#)' file and the '[assets/vocab](#)' folder in my repository.

