Computerized Source Criticism of Biblical Texts[1]

Idan Dershowitz,[2] Moshe Koppel,[3] Navot Akiva,[3] Nachum Dershowitz[4]

"We instruct the computer to ignore what we call grammatical words—articles, prepositions, pronouns, modal verbs, which have a high frequency rating in all discourse. Then we get to the real nitty-gritty, what we call the lexical words, the words that carry a distinctive semantic content. Words like love or dark or heart or God. Let's see." So he taps away on the keyboard and instantly my favourite word appears on the screen.

— David Lodge, *Small World* (1984)

Abstract

We have developed an automated method to separate biblical texts according to author or scribal school. At the core of this new approach is the identification of correlations in word preference that are then used to quantify stylistic similarity between sections. In so doing, our method ignores literary features—such as possible repetitions, narrative breaks, and contradictions—and focuses on the least subjective criterion employed by Bible scholars to identify signs of composition. The computerized system is unique in its ability to consider subtle stylistic preferences in aggregate, whereas human scholars are generally limited to cases where a word preference is pronounced. Our method is also less liable to accusations of bias, thanks to its reliance on context-independent criteria. Its efficacy is demonstrated in its successful deconstruction of an artificial book, *Jer-iel*, made up of randomly interleaved snippets from Jeremiah and Ezekiel. When applied to Genesis–Numbers, the method divides the text into constituents that correlate closely with common notions of "Priestly" and "non-Priestly" material. No such corroboration is forthcoming for the classic Yahwistic/Elohistic division.

---

[1]We are grateful to Drs. Shimon Gesundheit and Avi Shmidman for their valuable assistance and suggestions.

[2] Department of Bible, Hebrew University.

[3] Department of Computer Science, Bar-Ilan University.

[4] School of Computer Science, Tel Aviv University.

1 Introduction

In this paper, we introduce a novel computerized method for source analysis of biblical texts.

The matter of the Pentateuch's composition has been the subject of some controversy in modern times. From the late 19th century until recent years, the Documentary Hypothesis was the most prevalent model among Bible scholars. Since then, scholars have increasingly called into question the existence of some or all of the postulated documents. Many prefer a supplementary model to a documentary one, while others believe the text to be an amalgam of numerous fragments. The closest thing to a consensus today—and it too has its detractors—is that there exists a certain meaningful dichotomy between Priestly (P) and non-Priestly texts.[5]

The various source analyses that have been proposed to date are based on a combination of literary, historical, and linguistic evidence.

Our research is a first attempt to put source analysis on as empirical a footing as possible by marshaling the most recent methods in computational linguistics. The strength of this approach lies in its "robotic" objectivity and rigor. Its weakness is that it is limited to certain linguistic features and doesn't take into account any literary or historical considerations.

Though this work does not address the question of editorial model, we do hope it might contribute to the fundamental issue of literary origins. For cases in which scholars have an idea how many primary components are present, our new algorithmic method can disentangle the text with a high degree of confidence.

The method is a variation on one traditionally employed by the Bible scholar, namely, word preference. Synonym choice can be useful in identifying schools of authors, as well as individuals. However, despite their great utility, occurrences in the text of any one of a set of synonyms are relatively sparse. Therefore, synonyms are useful for teasing out some textual units, but not all. Accordingly, we use a two-stage process. We first find a reliable partial source division based on synonym usage. (Only a preference of one term over its alternative is registered; the context in which it is used is ignored.) In the second stage, we analyze this initial division for more general lexical preferences and extrapolate from these to obtain a more complete and fine-grained source division.

As noted, the advantage of a numerical lexical approach is its fundamentally objective nature. While potentially valuable, literary observations and historical reconstructions are particularly prone to controversy. For instance, a repetition may be viewed by one scholar as a telltale sign of multiple sources and by another as an intentional literary device. While our computerized method is objective and powerful, the narrow focus

---

[5] For a review of the current landscape, see Konrad Schmid, "Has European Scholarship Abandoned the Documentary Hypothesis? Some Reminders on its History and Remarks on its Current Status," in *The Pentateuch: International Perspectives on Current Research* (ed. Thomas B. Dozeman, Konrad Schmid, and Baruch J. Schwartz; *Forschungen zum Alten Testament* 78; Tübingen: Mohr Siebeck, 2011), 17–30.

on lexical analysis can occasionally lead to anomalous assignments of provenance that elementary non-lexical considerations (ideology, narrative consistency, repetitions, continuity, etc.) would have precluded.

The algorithm is generic in that it can be applied to any collection of biblical texts (or, for that matter, to other corpora). Other than the consonantal text, the only information used is Strong's Concordance, for the purpose of sense disambiguation and synonymy.[6] No prior knowledge regarding authorship is required. Thus, we will confirm the overall effectiveness of our method by testing it on an *artificial* book, *Jer-iel,* constructed by randomly interweaving the books of Jeremiah and Ezekiel. The algorithm is indeed able to separate this *Jer-iel* composite into its constituents with extremely high accuracy: 96%, as described in detail below. Moreover, when our automated methods are applied to the first four books of Moses, we will see that the results largely correspond to the "consensus" Priestly/non-Priestly (P/non-P) dichotomy (with some notable exceptions, which we discuss below). This suggests that our method may provide a powerful new instrument for the scholar's toolbox.

2 Previous Work

Author attribution is an active area of computer-science research. In the standard problem, which is not the one addressed herein, there is a known list of potential authors for each of whom there are sample writings. Then, the task of the algorithm is to apply automated computational methods to determine who among those is the true author of some anonymous text. Current methods achieve this goal by comparing quantifiable characteristic features of the unknown work to the sample writings. This is called the *authorship attribution* problem.[7] Another problem—closer in spirit to the task at hand, though not identical—is the *author clustering* problem. In this problem, one seeks to divide a collection of writings into a predetermined number of clusters, each written by a distinct author, by identifying shared measurable commonalities among the given works.[8]

What we attempt here is to take a single text and segment it along authorial boundaries and only then cluster the derived segments. This differs from clustering where one begins with single-author units and then simply assigns each such unit to the appropriate cluster. Surprisingly little work has been done in computer science to date on automatically identifying multiple authors within a single text, although some re-

---

[6] James Strong, *The Exhaustive Concordance of the Bible* (Nashville 1890). We made use of an online edition: http://www.htmlbible.com/sacrednamebiblecom/kjvstrongs/STRINDEX.htm (accessed August 23, 2013).

[7] Patrick Juola, *Authorship Attribution* (Now Publishers: Delft, 2008); Moshe Koppel, Jonathan Schler, and Shlomo Argamon, "Computational Methods in Authorship Attribution," *Journal of the American Society for Information Science and Technology*, 60 (2009): 9–26; Efstathios Stamatatos, "A Survey of Modern Authorship Attribution Methods," *Journal of the American Society for Information Science and Technology*, 60 (2009): 538–556.

[8] Robert Layton, Paul Watters, and Richard Dazeley, "Automated Unsupervised Authorship Analysis Using Evidence Accumulation Clustering," *Natural Language Engineering* 19 (2013): 95–120.

search has been done on identifying language and register transitions.[9] Those who have addressed similar questions, such as plagiarism detection, either assume that there is a single dominant author or that some pairs of units labeled as same-author/different-author are available for training purposes.[10] We make no such assumptions.

Several studies consider the problem of identifying the author of disputed New Testament books from among a set of known biblical authors.[11] As noted above, this classification problem is quite distinct from the biblical decomposition problem we are considering, where one text needs to be separated into authorial strands. Other, earlier computational works on biblical authorship questions use various methods to test whether the clusters in a given clustering of some biblical text are sufficiently distinct to be regarded as a composite text.[12] However, it is a simple matter to find some significant differences or similarities between two texts and to point to these as indicative of separate or identical sources. Such arguments are, therefore, unconvincing unless it can be shown that observed differences can be exploited to provide the correct split in cases where ground truth is known.[13] This is what we proceed to do next.

3 Synonym-Based Source Division

In this section, we describe an algorithm for automated clustering of single-author textual units, preliminary to the full-fledged source-division method of the next section. By way of illustration, we take the 52 chapters of Jeremiah and 48 chapters of Ezekiel, two roughly contemporaneous prophetic books, as our corpus. Given their

---

[9] An early algorithmic approach is Aravind K. Joshi, "Processing of Sentences with Intra-sentential Code-Switching," in *Proceedings of the 9th Conference on Computational Linguistics* (ed. Ján Horecký; North-Holland Linguistic Series 47; Prague: Academia, 1982), 145–150. See Donald Winford, *An Introduction to Contact Linguistics* (Malden, MA: Blackwell Publishing, 2003), 126–167.

[10] For the former, see, e.g., Sven Meyer zu Eissen and Benno Stein, "Intrinsic Plagiarism Detection," in *Advances in Information Retrieval: Proceedings of the 28th European Conference on IR Research* (ed. Mounia Lalmas, et al.; Logic in Computer Science 3936; London: Springer, 2006), 565–569; David Guthrie, Louise Guthrie, and Yorick Wilks, "An Unsupervised Probabilistic Approach for the Detection of Outliers in Corpora," in *Proceedings of the Sixth International Language Resources and Evaluation Conference,* (Paris: European Language Resources Association, 2008), 28–30. For the latter, see, e.g. Neil Graham, Graeme Hirst, and Bhaskara Marthi, "Segmenting Documents by Stylistic Character," *Natural Language Engineering* 11 (2005): 397–415.

[11] David L. Mealand, "Correspondence Analysis of Luke," *Literary and Linguistic Computing* 10 (1995): 171–182; Matthew J. Berryman, Andrew Allison, and Derek Abbott, "Statistical Techniques for Text Classification Based on Word Recurrence Intervals," *Fluctuation and Noise Letters* 3 (2003): 1–10.

[12] Yehuda T. Radday, "Isaiah and the Computer: A Preliminary Report," *Computers and the Humanities* 5 (1970): 65–73; Ronald E. Bee, "Statistical Methods in the Study of the Masoretic Text of the Old Testament," *Journal of the Royal Statistical Society* 134 (1971): 611–622; David I. Holmes, "Authorship Attribution," *Computers and the Humanities* 28 (1994): 87–106.

[13] A. Dean Forbes, "A Critique of Statistical Approaches to the Isaiah Authorship Problem," in *Association Internationale Bible et Informatique, Actes du Troisième Colloque International* (Paris: Champion 1992), 531–545.

100 unlabeled, unordered chapters, the task of the algorithm is to separate them out into the two constituent books.[14]

3.1 Stage 1: Initial Clustering

To obtain a word-based source division, we first employ one of the key features often used to classify different components of biblical literature, namely, synonym choice. The underlying hypothesis is that different authorial components are likely to differ in the proportions with which alternative words from a synonym set (henceforth: synset) are used. As is well known, this hypothesis has played a part in the critical analysis of the Bible since the pioneering work of Astruc who used a single synonym set—divine names—to divide the book of Genesis.[15] For our purposes, we regard occurrences of distinct words to be "synonymous" if they are identically translated in the King James Version. For example, the translations of both נטעתיו (lexical form: נטע) and אשתלנו (lexical form: שתל) include the English lemma "plant," and are thus treated as synonyms. It is not necessary for the terms to be identical in nuance (if such a thing exists); rough equivalence in usage is sufficient.[16] This definition yields 517 synsets in the Hebrew Bible, comprising a total of 1551 individual terms. Most sets consist of only two terms, but some include many more. For example, there are seven Hebrew words corresponding to "fear."

With these synsets in hand, we can obtain a measure of similarity between any two chapters. Whenever both chapters use words from the same synset, we look to see whether the choice of term is the same or different. The greater the proportion of sets for which the choices are the same, the greater the measure of similarity.[17] Specialized algorithms are then used to cluster the chapters, so that those in the same cluster are as similar as possible, while chapters in distinct clusters are as dissimilar as possible.[18]

---

[14] In principle, the clustering algorithm could create any number of clusters, corresponding to any given number of authors; for this example, we take it as given that the correct number of authorial clusters is two.

[15] Jean Astruc (published anonymously), *Conjectures sur les mémoires originaux dont il paroit que Moyse s'est servi pour composer le livre de la Genèse* (Brussels 1753).

[16] We manually deleted obvious mistakes, such as unrelated words that are translated to different senses of the same English word. We also merged synsets containing the same word in overlapping senses, including the set of divine names. It is perhaps noteworthy that this synset rarely affects results. (See the discussion of Genesis 1 below for a possible exception.)

[17] Formally, we adapt a similarity measure known as *cosine* (after its analogous use for capturing the magnitude of angles). For details, see Moshe Koppel, Navot Akiva, Idan Dershowitz, and Nachum Dershowitz, "Unsupervised Decomposition of a Document into Authorial Components," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Portland: Association for Computational Linguistics), 1356–1364.

[18] Specifically, we regard each chapter as a node in a graph each edge of which is weighted according to the similarity of the nodes it connects. We then seek a "minimal cut" of the graph into connected components, one component per cluster. Details of the method can be found in Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis, "Weighted Graph Cuts without Eigenvectors: A Multilevel Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (2007): 1944–1957.

We apply this method to separate the 100 chapters of Jeremiah and Ezekiel into precisely two clusters. The result is one cluster of 53 chapters, of which 48 are Jeremiah, and another cluster, with 47 chapters, of which 43 are from Ezekiel. We may purchase intuition for the process by considering some concrete examples. There are two Hebrew terms (פאה and מקצוע), corresponding to the English word "corner"; two (מנחה and תרומה) corresponding to the word "oblation"; and two (נטע and שתל) corresponding to the word "planted." We find that three choices (פאה ,מנחה, and נטע) tend to be located in the same units and, concomitantly, the other three alternatives (מקצוע ,תרומה, and שתל) are located in other units; the former are all Jeremiah and the latter are all Ezekiel. The algorithm takes into consideration the combination of synonym choices made in each chapter.

This synonym-based clustering is fairly good, but we can do much better. We observe that some chapters are assigned to one cluster or the other only because it is the nature of the clustering algorithm to classify every unit, however weak the evidence. But in fact borderline chapters may have only a weak affinity to other chapters in that cluster and are not part of what we might think of as its "core" components. We can compute a "center of gravity" of each cluster and use proximity to it as a basis for identifying the core chapters of each of the two clusters. When we formally compute cores in our Jeremiah-Ezekiel experiment, we are left with 74 chapters that split between Jeremiah and Ezekiel with only two misplaced units. Thus, we have a much better clustering, even if only a partial one.

3.2 Stage Two: Learning a Chapter Classifier

Now that we have what appear to be strong representative units for each author, we can use them to classify the remaining unclustered, non-core chapters. Recall that the classification task, in which we wish to assign an anonymous text to one of several potential authors (for whom we have writing samples), is well understood. By analyzing common features of our core chapters, we can automatically formulate rules that best characterize the differences between authors; these rules, aggregately known as a *classifier*, are then used to classify all chapters. Roughly speaking, the method that we use for finding such a classifier assigns to each textual feature some weight in support of one class or the other.[19] A text is then assigned by the classifier to the class with greater aggregate weight.

The learned rules depend, of course, on the choice of the types of textual features that we are considering. In general, it is known that the best features for textual classification are simply the frequency of use of each word that appears often enough in the corpus. The algorithm finds a variety of words used differentially by the two presumed authors.[20] For example, על ,הזאת, and הזה are over-represented in one of the cluster cores (the one corresponding mainly to Jeremiah), while אדם ,והנה, and אחד are

---

[19] We use support vector machines, as described in Corinna Cortes and Vladimir Vapnik, "Support-Vector Networks," *Machine Learning* 20 (1995): 273–297.

[20] Our definition of "word" here is a series of uninterrupted letters—not necessarily a single lexeme.

over-represented in the other cluster core. Hence, these words will be assigned considerable weight in support of the respective classes in which they are frequent.[21]

We use our learning algorithm to learn a classifier based on our core chapters. This classifier is then used to classify all chapters, including the other, non-core, chapters. The result is remarkable: we obtain a near-perfect split of the 100 chapters. Even the two Ezekiel chapters that were previously in the Jeremiah core fall to the Ezekiel side; only Ezekiel 42 is incorrectly classified.

3.3 Testing the Method

To further establish the efficacy of our method, we introduce Isaiah 1–33 into the mix. From among these three books we have three pairs of books: Jeremiah|Ezekiel, Jeremiah|Isaiah, Ezekiel|Isaiah. For each of the book pairs, the algorithm is given all the chapters in the two books but no information regarding which chapter came from which book, and we ask the algorithm to cluster the chapters of the two books as best as possible. We find that our two-stage method achieves near perfect results for those prophets (94–99%). Moreover, the chapters of each of the three prophetic works are automatically sorted from the chapters of a book in a different genre (Job) with not a single mistake.

4 Artificially Mixed Books

Up until now, we have considered the case where we are given text that is pre-segmented into chapters, each of which is known to be from a single book. This does not capture the kind of decomposition problems faced in the Pentateuch, where the text is divided into chapters, but there is no necessary correlation between chapter breaks and crossovers between authorial styles. Thus, we wish now to generalize our two-stage method to handle unsegmented text.

To make the challenge precise, consider how we might artificially create the kind of document that we wish to decompose into sources. We create a composite document, called *Jer-iel*, by first choosing a random number of contiguous verses (between 1 and 100) from the beginning of Jeremiah, then some random quantity from the beginning of Ezekiel, then some from the remaining verses of Jeremiah, and so on until one of the books is exhausted, at which point we take the remaining verses of the other book. We wish to find an algorithm that, given no information beyond the composite document itself, can split the verses of the composite document into two sets, ideally with one consisting of Jeremiah and the other of Ezekiel.

We adapt the two-stage method described above in the following way. First, we "chunk" the text into segments of 40 verses each, in order to create artificial "chap-

---

[21] The assigned weights for these and a few other relatively significant words are as follows: כי 15.0, הזה 10.9, על 6.2, גם 4.9, הזאת 4.7, אחד -0.8, וכל -3.2, והנה -4.1, אדם -12.6, אני -14.9. Positive weights are indicative of Jeremiah, and negative weights point to Ezekiel.

ters." Of course, each such segment is not homogeneous and is likely to include verses from both Jeremiah and Ezekiel. In fact, in our composite *Jer-iel*, about 30% of the segments are mixed and the rest are either pure Jeremiah or pure Ezekiel.

We then run the two-stage method on the segmented text, just as we did on the homogeneous chapters above. Specifically, we encode the segments as lists of synonym occurrences and cluster them. Then we identify the cores of each cluster. The key is for the cores to consist primarily of pure segments.

In fact, when we apply this part of our algorithm to *Jer-iel*, we find that all of the pure Jeremiah segments are in one core and all the Ezekiel segments are in another core. In addition, there are some mixed segments in each core.

Following our algorithm as described above, we now use these cores to learn a classifier that automatically identifies *verses* as more similar to the first cluster core or more similar to the second cluster core. Unlike for the earlier case, at this stage, we are not classifying whole segments (which might be of mixed origin), but rather individual verses. This allows us to obtain a fine-grained division of the text.

The problem with classifying individual verses is that they are short and may contain few or no characteristic features of either book. To remedy this, and also to take advantage of the "stickiness" of classes across consecutive verses (if a given verse is from a certain book, there is a good chance that the next verse is from the same book), we use a "smoothing" procedure. If a verse is not strongly assigned to either class, we check the class of the last assigned verse before it and the first assigned verse after it. If these are the same, the verse is assigned to that class. If they are not, we determine some optimal split point using a formal method and assign all verses before that point to the same class as the last assigned prior verse and all verses following that point to the same class as the first assigned subsequent verse. (In broad terms, we choose the split point that makes the verses on its two sides as different as possible.)

Employing this optional procedure generally gives somewhat stronger overall results, but it papers over certain very interesting observations regarding small pockets of verses that are not from the same class as surrounding verses. This phenomenon will become more apparent below, when we consider the division of the first four books of the Pentateuch.

Using our method, with smoothing, on *Jer-iel*, we obtain a split of verses in which 96% of the verses are correctly assigned.

When applying the same method to other same-genre merged books, we obtain 83% accuracy for Isaiah-Jeremiah and 88% accuracy for Isaiah-Ezekiel. When we create merged books by combining each of the three prophetic works with Job, we are able to sort out the verses with accuracy ranging from 89% to 95%. In other words, our algorithm can successfully tease apart components of an artificially merged document, whether of the same genre or not, with quite high accuracy.

5 Automated Source Division of the Pentateuch

Having demonstrated the efficacy of our method, we wish to apply it to the Pentateuch. While there is little agreement among Bible scholars regarding the composition of the Pentateuch, there exists a common denominator among most experts: the first four books are made up of Priestly and non-Priestly material. We therefore endeavor to find the optimal binary split of Genesis–Numbers.[22]

We now show intermediate results for our method as it progresses through each step of the algorithm.

5.1 Stage 1: Initial Clustering

Initially, we encode each chapter in terms of the synonyms used (or not used) in that chapter for each of the synsets, as described above. We then measure the difference/similarity of every pair of chapters and use this information to cluster the chapters into two clusters. (Note that this is a crude clustering in which chapters are treated as coherent units; in the next stage, we drop this assumption.) We then consider the cores of each of these two clusters.

At this stage, we obtain the following two cores:

Cluster Core 1 (53 chapters)

Exodus 16, 25, 28–31, 36–39

Leviticus 1–12, 14, 16, 19, 22–25, 27

Numbers 1–3, 5–10, 13–15, 17–19, 26–29, 31, 33–34, 36

Cluster Core 2 (37 chapters)

Genesis 4–5, 12–13, 16, 18, 21,26, 29–34, 36–37, 41–47, 50

Exodus 2, 3, 8, 17–19, 22–24, 34

Numbers 21–23

It is already clear that Cluster 1 corresponds roughly with Priestly (P) sections, and Cluster 2 with non-Priestly sections. Since at this stage we treat chapters as though they were coherent units, a number of mixed chapters are assigned to one cluster or another.

---

[22] We obtain similar results and nearly identical accuracy levels when applying the method to all five books of the Pentateuch. However, given that there is little agreement regarding the boundaries of the edited composition, and since our experiment is a binary split, we opted to examine Genesis through Numbers for the purpose of this article. It may be interesting to run similar experiments in the future with different bounds: Gen–Josh, Deut–2 Kgs, etc. We omitted poetry from our analysis, due to its distinct language register.

In Table 1, we show synonym choices that characterize the respective cluster cores. For each synonym, we show in the left column the percentage of chapters of the P core in which the synonym appears, and in the right we show the same for the non-P core.

We note that the two cluster cores make consistent lexical choices over a number of apparently unrelated synsets. In some cases, such as names of God, there are synonyms that are used in both clusters, while others are used only in one of the clusters.[23] In other cases, such as בגד and שמלה, each cluster is characterized by a particular choice of synonym. Thus, for example, we have וכבס בגדיו (Leviticus 13:6) but וכבסו שמלתם (Exodus 19:10).

Most of the distinguishing terms picked up by our algorithm are well known among Bible scholars.[24] The less pronounced synsets, however—אמר/דבר, for instance—are not widely appreciated. This is to be expected, as a weak predilection of a source in one direction or the other requires precise counts of word occurrences—something better left to a machine. Our method's ability to grapple with subtle tendencies of this nature is one of its most salient advantages.

5.2 Stage 2: Learning a Verse Classifier

All the above core chapters are now used as the basis for a second round of classification. The method automatically identifies relatively frequent words (not necessarily in our synsets and often made up of multiple undivided lexemes) that are found with widely differing frequencies in the two cluster cores. These words are used to construct the best possible classifier for distinguishing Type 1 (P) texts from Type 2 (non-P) texts, and this classifier is used to classify *every* individual verse in Genesis–Numbers as one type or the other.

As described in Section 3.2 above, the computed classifier assigns "weights" to words, thus designating them as markers of one class or the other. In Table 2, we list the thirty words to which the classifier assigned the most weight for each of the respective classes. For each word, we show in the left column the weight given by the classifier to each occurrence of the word, with positive weights for P and negative weights for non-P. The middle column shows the frequency of the word's occurrence relative to the total number of words in the P core chapters, and the right column gives the same for the non-P core.

We make three primary observations about this list. First, we note that it is interesting that an initial clustering based solely on synonym choice, with no thematic criteria, should result in a division that so clearly splits the text along certain thematic lines.

---

[23] Note that with regard to names of God, our method does not take advantage of potentially relevant thematic material, such as Exodus 6.

[24] See, e.g., Joseph Estlin Carpenter, and George Hartford-Battersby, *The Hexateuch: According to the Revised Version*, vol. 1 (London: Longmans, Green & Co. 1900), 185–221; Heinrich Holzinger, *Einleitung in den Hexateuch* (Freiburg: Mohr Siebeck, 1893), 93–110, 181–191, 283–291, 338–349.

Most conspicuously, of course, the word הכהן appears quite frequently in the core chapters of what turns out to be the P cluster but not at all in the non-P core. Second, it is noteworthy that the Tetragrammaton is frequent in both cores, but its *absence* turns out to be a marker of the non-P cluster; though not obvious from this table, there are virtually no core P chapters in which it does not appear.[25] Third, we find that clustering according to synonym choices reveals yet other differences in lexical choice regarding words that are not obviously related to the synsets considered earlier. Thus, unexpectedly, we find that the function word עליו appears four times as often in the P core, while לי occurs four times as often in the non-P core.

Of course, this is only a partial list, and, while numerous distinguishing words familiar to Bible scholars are present, others are found further down the full list. Sometimes, this is because the word tends to co-occur with other significant words, and therefore each of the co-occurring words is assigned a lower weight. Other times, it is because the word's distribution across the chapters in the class is not sufficiently uniform.

Many other interesting words are found just slightly further down the full list and are worth mentioning. For instance, words that contribute to the assignment of a verse as P include לכל, ואל, במים, צבא, and שקל, while חן, ארצה, מאד, הנער, ותאמר, and גדול, among others, contribute to a non-P classification.

More interesting, perhaps, are the "generic" terms, like ויאמר, which is eight times more likely to appear in the non-P core than in P (a similar ratio holds for non-core verses), versus וידבר, which is five times more likely in P. The algorithm takes quite subtle differences into account, too; for example, אם, which is only slightly less common in the P core, is assigned a non-negligible weight by the classifier. The combination of all these features—strong markers and weaker ones, each with the appropriate weight—is what gives the classifier its discriminative strength.

In the next step, we use our learning algorithm to establish a classifier that is then used to categorize each verse in Genesis–Numbers. After smoothing (explained above), the algorithm proposes the following split of verses:[26]

P

Genesis 9:18–10:31, 15:18–16:1, 19:23–27, 22:21–23:20, 25:1–18, 34:24–30, 35:20–36:39
Exodus 5:13–21, 6:4–7:8, 9:4–7, 12:2–28, 12:40–13:1, 13:21–14:3, 14:8–10, 14:27–16:36, 20:9–17, 24:2–31:17, 34:3–7, 34:21–40:38
Leviticus 1:1–25:6, 25:30–27:34
Numbers 1:1–10:28, 10:33–11:7, 12:16–13:26, 13:32–14:10, 14:25–20:25, 25:5–33:56

---

[25] Likewise the plural construct form בני. Similarly, כי and ויאמר appear in almost all core chapters of the non-P cluster.

[26] The full results prior to smoothing are too long to include here but can be found online at http://nachum.org/summary.html and http://nachum.org/results.html.

Non-P

Genesis 1:1–9:17, 10:32–15:17, 16:2–19:22, 19:28–22:20, 24, 25:19–34:23, 34:31–35:19, 36:40–50:26
Exodus 1:1–5:12, 5:22–6:3, 7:9–9:3, 9:8–12:1, 12:29–39, 13:2–20, 14:4–7, 14:11–26, 17:1–20:8, 20:18–24:1, 31:18–34:2, 34:8–20
Leviticus 25:7–29
Numbers 10:29–32, 11:8–12:15, 13:27–31, 14:11–24, 20:26–25:4

Now we wish to compare the smoothed results with those obtained by scholars using traditional methods. We use Nöldeke's seminal source analysis as our initial point of reference.[27] As can be seen in the table, the results correspond quite closely with Nöldeke's P/non-P division. To be precise, our method's split (after smoothing) aligns with Nöldeke for 86.6% of the verses. While this figure is already noteworthy, it turns out that despite the comparative resilience of Nöldeke's analysis, our algorithm has a tendency to disagree with his classification specifically where it deviates from the subsequent majority opinion. For instance, Nöldeke takes Exodus 17 to be Priestly, whereas our method classifies it as non-P. As it happens, nearly all Bible scholars agree that the bulk of that chapter is non-P. Similar examples abound. Therefore, we compare the algorithm's results against both Nöldeke's division and a "consensus" of various scholars, which we use as our primary benchmark.[28] We find that for those verses for which all these scholars agree, the algorithm's split corresponds with the consensus split for 91.4% of the verses.[29]

We visually display the correspondence between the respective divisions in Figure 1. Each of the "barcodes" represents a division of the text. A horizontal line represents a single verse, the first verse of Genesis lying at the top and the last verse in Numbers lying at the bottom. A line is green if the corresponding verse is assigned to P and red if the verse is assigned to non-P. Gray indicates that there is no consensus.

6 Discussion

It might be noted that our method's split corresponds to a considerable extent with that between narrative and legal sections of the Pentateuch. In fact, many of the sections for which our method's split does not correspond with the benchmark P/non-P

---

[27] Theodor Nöldeke, *UntersuchunGenesis Zur Kritik Des Alten Testaments* (Kiel: Schwer-s'sche Buchhandlung, 1869), 143–144. Nöldeke of course uses different terminology.

[28] It would be impossible—technically and fundamentally—to establish a true consensus opinion vis-à-vis Pentateuchal source analysis. In addition to Nöldeke, we currently include the analyses of Samuel Rolles Driver, *An Introduction to the Literature of the Old Testament* (9th ed.; Edinburgh: T&T Clark, 1913), and Richard Elliott Friedman, *The Bible with Sources Revealed* (San Francisco: HarperCollins, 2003). The selection of these scholars was dictated largely by accessibility; we intend to gradually update our benchmark to account for as many opinions as possible.

[29] Without smoothing, the method obtains a correspondence of 85.2%.

split are narrative sections in P and legal sections in non-P.[30] Nevertheless, there are numerous examples where our method's split corresponds perfectly with the benchmark P/non-P split within a narrative section or, alternatively, within a legal section.

To appreciate this point, let us consider in some detail Numbers 13–16, where there are numerous transitions between legal and narrative sections. At Numbers 13:32, the return of the spies to the desert, there begins a string of verses assigned by our method, as by the benchmark division, to P. It is important to recall that our method's assignments are based on an optimal aggregation of small bits of evidence. Thus, these verses are assigned to P because of the presence of words such as בני and the Tetragrammaton, both highly weighted for P. As in the benchmark division, our method finds a transition to non-P at 14:11. The indicators of non-P in this section, a dialogue between Moses and God that extends through 14:24, are נא, העם, כי, and ויאמר. Numbers 14:25–38, God speaking to Moses and Aaron, is assigned, approximately as in the scholarly benchmark, to P, the main indicators being בני, לכל, לכם, and the Tetragrammaton.

The algorithm assigns Numbers 15 to P as in the benchmark. Numbers 16, the story of Korah, is a narrative section that is regarded by many to be an amalgamation of (perhaps multiple strata of) P and non-P. Our smoothed method assigns the chapter to P, but the unsmoothed method tells a more nuanced story. Among the strong P words in this section, we find לכם, לפני, ואל, לכל, and the Tetragrammaton. Among the strong non-P words, we find נא, מאד, לנו, גם, כי, and אם. Overall, verses 1–11 and 16–25 are assigned to P and verses 12–15 and 26–30 assigned to non-P, corresponding almost perfectly to the benchmark—despite the fact that our considerations are entirely lexical and we do not take into account thematic or ideological considerations at all. One point of interest may be verse 16:33. Whereas some scholars see marks of P or redaction in the previous verse, verse 33 is generally considered to be wholly non-P. Our unsmoothed method, however, clusters the verse with P, due in part to the words קהל and תוך, which appear at the end of the verse. Given the possibly fragmentary account of Korah's death in P, with some attributing the orphaned passage ואת כל האדם אשר לקרח ואת כל הרכוש to that source (or a harmonizing editor), it may in fact be worth considering the possibility that the phrase ויאבדו מתוך הקהל is external to the non-P narrative.

The most prominent case in which our division departs from the benchmark opinion is Genesis 1:1–2:4a. Our method places the section in the predominantly non-P cluster, despite Bible scholars' nearly unanimous agreement that it is Priestly in origin. Our divergent results, in this case, can perhaps be attributed to a few factors that conspire to mislead the computer algorithm. One issue is the prevalence of the word א-להים, which our method considers to be a strong indicator of non-Priestliness, as is quite evident in Table 2. As noted above, our method doesn't account for any transition in Exodus 6 (P). A second factor is the repeated appearance of the verb ויאמר. This word is also associated with the non-Priestly texts, since P generally prefers the verb וידבר, as mentioned above. However, ויאמר in Genesis 1 does not take an indirect object,

[30] Perhaps unsurprisingly, P passages generally ascribed to the Holiness stratum are disproportionately likely to be classified by our method as non-P.

whereas in the vast majority of cases in which P opts for וידבר, it is followed by אֶל, etc. Therefore, ויאמר here is not truly indicative of non-P, but our method, which is blind to syntax, cannot distinguish between these two usages of ויאמר.[31] All that being said, the terms that the method identifies as markedly non-P in this section—the aforementioned ויאמר and א-להים, along with כי—feature in the narrative's so-called *Wortbericht* elements, which some scholars indeed believe to pre-Priestly, or perhaps more accurately, pre-P^{g}.[32] It may therefore be worth considering our method's results in light of the questions surrounding the literary history of Genesis 1:1–2:4a.

Another interesting example is Genesis 9. Though our method agrees with many scholars that there is a source change at verse 18, the attribution of the two sources is reversed. Whereas most scholars assign the first part of the chapter to P, our method clusters that section with non-P.

There are several cases in which our method's division may have something to contribute to the scholarly conversation. One such example is Genesis 18:17–19. Nöldeke considers the passage to be non-Priestly (as do the other scholars), while the unsmoothed version of our method clusters it with P.[33] These three verses were attributed by Wellhausen to a late supplementer, due to perceived similarities with Genesis 13:14–17 and 22:15–18—both of which he considered to be redactional—as well as "suspicious language."[34] The status of the passage, together with its broader context, is debated to this day.[35] The fact that our method finds that the three verses stand out from their surroundings indicates that the language is indeed suspicious. The confluence of the words אשר, על, and יהיה—each of which is more prevalent in P than in non-P—is particularly notable in this context. In addition to voicing an opinion regarding debated passages, our method may thus prove useful as a tool for identifying areas worthy of further study wherever it disagrees with scholars in the field.[36]

We have seen thus far that there is a high degree of correspondence between our algorithm's results and previously proposed P versus non-P source divisions. However, when we use the same synonym-driven method to break up the text into three categories, the non-P verses (in the first four books) do not split into anything like the classical J and E sources. Even if we try simply to split the non-P material into two, the resulting sub-clusters—and even their cores—do not correspond to the classical

---

[31] As we note below, we plan to incorporate syntactical and morphological data in a future version of our method.

[32] See Jürg Hutzli, "Tradition and Interpretation in Genesis 1:1–2:4a," *Journal of Hebrew Scriptures* 10 (2010), article 12. Regarding the status of the approbation formula, see pp. 19–20.

[33] We refer here to our unsmoothed results, which are pertinent for questions regarding individual verses. The smoothed results are better suited for identifying broader patterns.

[34] Julius Wellhausen, *Die Composition des Hexateuchs und der historischen Bücher des Alten Testaments* (3d ed.; Berlin: Georg Reimer, 1899), 26.

[35] See, e.g., Johannes Unsok Ro, "The Theological Concept of YHWH's Punitive Justice in the Hebrew Bible: Historical Development in the Context of the Judean Community in the Persian Period," *Vetus Testamentum* 61 (2011): 406–425.

[36] See, for instance, the brief discussion of Numbers 16:33b, above. We thank Prof. Jan Christian Gertz for highlighting this additional use for our method.

documents. There appear to be two possible explanations for this: (1) the J and E sources are not sufficiently distinct from one another in terms of word usage for our method to tease them apart; (2) the traditional J/E division is flawed. Either way, we find that while purely word-based classification is a useful new tool, it does not obviate philological analysis.

7 Conclusions

We have shown that documents can be deconstructed into authorial components with very high accuracy by using an automated two-stage process. First, we establish a reliable partial clustering of units by using synonym choice; next, we use these partial clusters as training texts to learn a text classifier using recurring words as features. The learned classifier is then used to classify verses in one authorial thread or another.

We have considered only decompositions into two components, although our method generalizes trivially to more than two components, for example by applying it iteratively. The real challenge is to determine the correct number of components, where this information is not given. We leave this for future work.

Despite this limitation, our success on artificially merged biblical books suggests that the method can be fruitfully applied to the Pentateuch, given that many scholars divide the text into two primary categories—Priestly and non-Priestly. We find that our algorithm's split corresponds to scholarly views regarding P and non-P for over 90% of verses.

Analysis of the disagreements suggests that our method is prone to assigning P narratives and H law to non-P. In some of these cases, such as Genesis 18:17–19 and Numbers 16:33b, our method may prove to have something to contribute to the discussion. Other times, such as Genesis 1:1–2:4a, the method's conclusions appear to conflict with the preponderance of evidence produced by other methods.[37]

Among the tools at our disposal for improving our method are the inclusion of measurable morphological and syntactical features (in addition to the lexical features we already use) and disambiguation of polysemic words (such as אל and לך) in the second phase of the algorithm. Likewise, some recent work of ours suggests that the first phase of the algorithm might be performed using binary lexical features (viz., the presence or absence of a common word in the text) rather than synonyms.[38]

For this paper, we exploited our new method to explore stylistic features in the Pentateuch. In many ways, this experiment has served as a proof of concept. We set out to establish the method's uniqueness and efficacy, rather than settle long-standing disputes in the field. In the future, we wish to provide more data regarding the strata and sources of the Pentateuch. We also look forward to applying our method to additional

---

[37] However, see discussion above for an alternative explanation of the discrepancy.

[38] Moshe Koppel and Navot Akiva, "A Generic Unsupervised Method for Decomposing Multi-Author Documents," *Journal of the Association for Information Science and Technology* 64 (2013): 2256–2264.

biblical books in the hope that it may shed some new light on unsettled questions of authorship.

Synonym		P Core	Non-P Core
captain	נשיא	19%	3%
	שר	2%	30%
clothes	בגד	19%	3%
	שמלה	0%	19%
earth	אדמה	0%	22%
	ארץ	47%	86%
go	בא	15%	14%
	הלך	8%	78%
God	א-ל	0%	19%
	א-להים	2%	57%
	י-הוה	96%	65%
man	אדם	30%	11%
	איש	21%	59%
manner	דבר	0%	8%
	משפט	9%	0%
meat	לחם	6%	0%
	אכל	2%	14%
midst/among	קרב	9%	16%
	תוך	49%	11%
near	נגש	6%	24%
	קרב	15%	11%
offer	הקריב	45%	0%
	זבח	4%	11%
put	נתן	49%	5%
	שת	0%	8%
	שם	17%	41%
said	אמר	85%	95%
	דבר	87%	59%
sin	חטאה	9%	14%
	חטא	11%	0%
south	נגב	9%	8%
	תימן	6%	0%
stone	סקל	0%	8%
	רגם	6%	0%

Table 1. Synonyms that characterize the cluster cores and the percentage of core chapters with that word choice.

Word	Weight	P	Non-P
הכהן	8.9	0.64%	0.00%
י-הוה	8.3	1.76%	1.04%
בני	7.3	1.32%	0.55%
ואת	6.7	1.44%	0.71%
אהרן	6.3	0.59%	0.06%
בן	6.0	0.71%	0.39%
לכם	5.8	0.44%	0.15%
זהב	5.6	0.39%	0.02%
לפני	5.5	0.54%	0.13%
העדה	5.4	0.18%	0.00%
אני	4.9	0.24%	0.11%
למטה	4.8	0.17%	0.00%
ויסעו	4.8	0.19%	0.04%
ישראל	4.7	0.98%	0.46%
עליו	4.7	0.32%	0.08%
צוה	4.7	0.29%	0.03%
ליהוה	4.4	0.70%	0.10%
משפחת	4.1	0.37%	0.01%
מועד	3.9	0.37%	0.00%
יהיה	3.9	0.37%	0.07%
קדש	3.9	0.34%	0.02%
וידבר	3.8	0.33%	0.07%
איש	3.8	0.38%	0.32%
ויחנו	3.8	0.17%	0.04%
המזבח	3.7	0.34%	0.01%
הקדש	3.6	0.26%	0.00%
הוא	3.5	0.51%	0.47%
אהל	3.5	0.31%	0.01%
המשכן	3.3	0.17%	0.00%
לאמר	3.3	0.46%	0.34%

Word	Weight	P	Non-P
ויאמר	-16.1	0.20%	1.70%
כי	-10.7	0.66%	1.70%
העם	-7.0	0.13%	0.39%
יעקב	-6.6	0.00%	0.64%
פרעה	-6.4	0.00%	0.57%
אם	-6.3	0.13%	0.15%
אלוף	-6.1	0.00%	0.20%
יוסף	-6.0	0.05%	0.71%
בארץ	-6.0	0.05%	0.43%
שם	-5.9	0.07%	0.43%
א-להים	-4.8	0.01%	0.46%
אברם	-4.7	0.00%	0.16%
ויקרא	-4.6	0.02%	0.29%
הא-להים	-4.3	0.00%	0.20%
לי	-4.3	0.11%	0.43%
ההר	-4.0	0.04%	0.13%
עשו	-3.9	0.06%	0.27%
נא	-3.8	0.01%	0.33%
מצרים	-3.5	0.11%	0.46%
ויבא	-3.5	0.02%	0.25%
גם	-3.4	0.02%	0.29%
אלי	-3.4	0.00%	0.22%
אנכי	-3.4	0.00%	0.25%
חמור	-3.4	0.00%	0.10%
הנה	-3.3	0.03%	0.29%
את	-3.1	3.10%	3.48%
ויהי	-3.0	0.08%	0.34%
לנו	-3.0	0.02%	0.17%
ישלם	-2.9	0.00%	0.08%
עם	-2.9	0.02%	0.20%

Table 2. Words to which our learned classifier assigned the highest weights for the respective classes, along with their frequencies (as a percentage of total words) in each class.

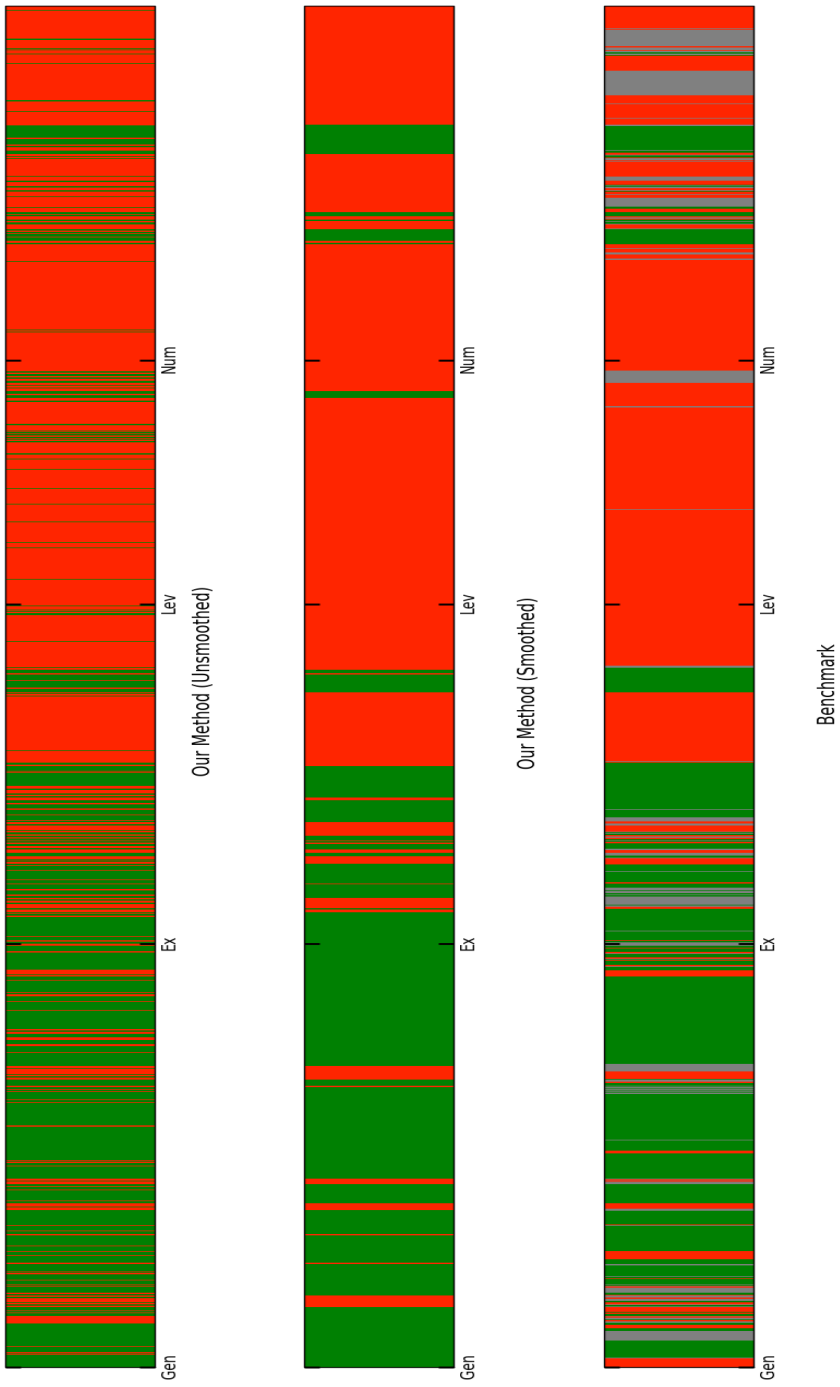

Figure 1. A visual display of the correspondence between our division—both unsmoothed (leftmost "barcode") and smoothed (center)—and the benchmark division (right).