Long Phrases in Torah Codes

Art Levitt[1], Nachum Bombach [2],Harold Gans [3], Robert Haralick [4], Leib Schwartzman [5], Chaim Stal[6]

[1] Computer Research Analyst, Jerusalem, Israel
[2] Certified Public Accountant and Rabbi, Jerusalem, Israel
[3] Cryptologist, Baltimore, Maryland, U.S.A.
[4] Computer Science Department, City University of New York, U.S.A.
[5] Mathematics and Pattern Recognition, Jerusalem, Israel
[6] Torah Scholar, Jerusalem, Israel

artlevitt23@yahoo.com

Abstract

The Torah code hypothesis states that the Torah (the first five books of the Hebrew Bible) contains within it letter sequences (codes) that were created intentionally, as a form of communication to human beings, the intended receivers. We test a long phrase aspect of the hypothesis, by first proposing a method for estimating the probability that letter sequences can be extracted from any given text, to form intelligible phrases simply by chance. We then apply this method to Torah code research, to a letter sequence extracted from the Torah, concerning bin Laden. This yields a p-level of 1.2e-05, the probability that this phrase would have occurred merely by chance.

1. Introduction

Torah code research is concerned with a particular type of letter sequence, formed by extracting equally spaced letters from a text. This is called an ELS (equidistant letter sequence). Of interest to the current study is a long ELS that consists of one or more phrases, which we call an "ELS phrase string" or simply a "string".

The letter extraction is done by ignoring all punctuation and inter-word spaces. For example, the string "tin tops" can be found starting with the first "t" in the word "punctuation" in the preceding sentence, and using a skip distance of +4 (that is, counting forward every 4 letters from the starting position).

According to the original Torah code hypothesis, logically or historically related words can be found as ELS's in the Torah, associated with each other significantly more often and in a more compact area than expected by chance. Many previous Torah code studies examined clustering of multiple ELS's by measuring their proximity to each other in a matrix [1]-[5].

The current study adds a simpler evaluation method to those already in use. It is concerned with evaluating a single ELS within a text, as in figure 1, rather than a

cluster of ELS's. It is therefore geometrically simpler; enabling the human reviewer to see the communication in a straight line (the code in the figure is in Hebrew, annotated with the English translation).

Figure 1: An ELS in the Torah of special interest

One of the most basic qualities of a communication is its intelligibility to the receiver. Under the null hypothesis of no Torah codes, we would expect that strings found in the Torah would be no more intelligible than those found in other (comparison) texts.

2. Description of the general method

2.1. Overview

Our method of significance estimation for an ELS phrase string found in a particular text is to compare its intelligibility to that of a large set of competitors. We do so by means of a large set of human reviewers who are asked to classify each string as to whether it is intelligible or not. With no indication of which string came from the original text, they classify it, along with the competitor strings extracted from a population of comparison texts. The popularity of a string is defined as the number of reviewers who classify it as intelligible. The relative popularity of the original string among all competitors determines its significance. A string - from any of these texts - is accepted for human review only if all of its words come from a lexicon of the language of interest.

2.2. Data preparation

2.2.1. The lexicon and the original string

The lexicon should cover as much of the language of interest as possible. The original ELS phrase string to be studied is typically found by starting with a particular keyword, called an "anchor". We seek any unusually long, intelligible ELS phrase strings containing that anchor and composed of words from the lexicon.

2.2.2. The comparison texts

A real text can be modified to create comparison texts. For example, from randomly permuted words, letters or passages of the real text, a comparison text can be constructed by computer. Such a comparison text is called a "monkey text" because of this randomness. In addition, an unmodified real text can be used. A random starting position and skip distance in such a text can be computer-selected. The text positions so defined can serve as the anchor for one trial, and the process can be repeated thousands of times.

2.2.3. Searching for competitor strings

To find ELS phrase strings in a comparison text, we exhaustively search all possible spacings of words, requiring only that they form a continuous ELS that includes the chosen anchor. A whole "tree" of possibilities may exist. For example: a string containing the letters "formedittoned" has at least two main branches, one branch after the word "for": ("for me", etc); and one after the word "form": ("form edit", etc); and further sub-branches exist as well. A competitor string must have a total length and an average word length that equal or exceed those of the original string.

2.3. Two review sessions

2.3.1. Collecting observed popularities

The competitor strings from the comparison texts are submitted to a large set of human reviewers, in two sessions (using two separate sets of reviewers). Under double blind protocol, the reviewers classify each listed string as "intelligible" or "not intelligible".

In session 1, each string is given to only one reviewer. Exceptions are: (1) the original string is mixed in with the competitors in a random, unmarked position of each reviewer's list; (2) a small set of control strings (manually created to appear to be intelligible) are mixed in, the same set included for each reviewer. As a requirement of acceptance of a reviewer's results, he or she must choose at least one of the controls, but not all of them. This avoids those who are very strict or very lax in accepting a string.

All strings chosen by all valid session 1 reviewers (including the original string) are gathered. This list is duplicated and sorted randomly for each session 2 reviewer. Therefore, each string accepted by a session 1 reviewer is judged by the full set of session 2 reviewers.

2.3.2. Deriving inherent popularity

Care is required to ensure that our popularity measure is a reasonable estimate of perceived intelligibility. We must account for the fact that each single session 1 reviewer acts as a kind of gatekeeper, and can prevent true competitors from reaching the session 2 review. Therefore the **observed popularity** trends - the session 2 results – must be refined. We are really interested in **inherent popularity** trends, the measure we would get if all strings were permitted to pass the gate.

We use a standard simulation technique to estimate inherent popularity. Our algorithm assigns a starting inherent popularity to each string, and subjects it to a simulated review 1, letting it pass the gate at a rate, or probability, dictated by the assigned inherent popularity. For example, if the inherent popularity level is set to 2 (out of say 22 simulated session 2 reviewers), this string has a chance of 1/11 of passing the gate in the simulation. If it passes the gate, it has a similar chance of being voted for by each simulated session 2 reviewer. Thousands of iterations are run, each slightly adjusting the inherent popularities, until arriving at the best fit to the actual session 2 observed popularities.

2.4. Obtaining a significance level

The p-level, **P**, for the experiment is:

$$P = s/t \tag{1}$$

where **s** is the number of non-control strings with higher inherent popularity than the original string, plus half the number of non-control strings with the same inherent popularity as the original string; and **t** is the number of accepted texts, estimated using the ratio of classified strings, as follows: **m** is defined as the total number of non-control strings given to all session 1 reviewers. **v** is the number of such strings actually classified by the valid reviewers (it excludes those that were skipped due to indecision or lack of time). **N** is the total number of texts used in the experiment. The estimate for **t** is:

$$t = (v/m)N \tag{2}$$

3. Case study

3.1. Description

Our case study (figure 1) is a particular ELS phrase string from the Torah translated as: **I will name you "Destruction". Cursed (is) bin Laden and revenge (belongs) to the Messiah**. We do not attempt to interpret this - only to gauge its intelligibility.

3.2. Results of data preparation

3.2.1. The lexicon

We build our lexicon from two sources, ancient and modern: (1) We use close to 40,000 words from the Hebrew Bible (we exclude the book of Daniel, since it contains many non-Hebrew [Aramaic] words). (2) We use all words from the online Hebrew news, Arutz-7, from the year 2002. This second source increases the lexicon size to almost 107,000 words.

3.2.2. The generated comparison texts

The comparison texts are created from two sources: (1) a population of 307,200 permuted Torah texts - 25,600 texts from each of 12 permutation methods: letter within word, verse, chapter, book, text; word within verse, chapter, book, text; and verse within chapter, book, text; (2) a "virtual" text population from a Bible text segment the same length as Torah. This segment begins immediately after the end of Torah (the book of Joshua), and continues until word 7 of Kings II, 18:24. From this segment we randomly pick the anchor location ("bin Laden") for each trial. We examine

approximately the same number of ELS's from both sources and therefore we consider that the number of texts examined (**N**) is 2 * 307,200 = 614,400.

3.2.3. The identified competitor strings

Surrounding every occurrence of the anchor in a comparison text, the computer searches for strings that have length at least equal to the original (29 letters) and average word length at least equal to the original (29/6, because we consider the anchor and its optional prefix letter to be one word). This yields **m** = 13,430 strings.

3.3. Results of review sessions

The 13,430 competitive strings are distributed so that each of 64 session 1 reviewers receive approximately 210 of them, plus 8 control strings and the original string randomly mixed in. 62 of the 64 students are valid, each accepting between 1 and 7 control strings as intelligible (interestingly, 41 of them accept the original string). They complete classifying a total of **v** = 12,880 non-control strings, assigning "intelligible" to 204 of them.

Table 1: Results for non-control strings

Popularity level	Number of Strings (Observed) *	Number of Strings (Inherent)
1	36	1331
2	16	150
3	9	39
4	3	9
5	2	5
6	2	4
7	3**	5**
8	0	0
9	0	0
10	1	1

* 133 non-control strings received 0 session 2 votes.
** Includes the original string

Each of 27 session 2 reviewers is given this full list of 204 strings, uniquely sorted, plus the 8 controls and the original string mixed in. 22 of the session 2 reviewers are valid, each selecting between 1 and 7 controls. Three of the controls receive more "votes" for intelligibility than the original string. Results for the non-control strings are in Table 1.

. The table lists the observed values, along with the estimated inherent values from a simulation, run as proposed in section 2.3.2. The simulation agrees with a logical assessment - that at a lower inherent popularity,

there is a larger excess of strings, due to the higher difficulty of being accepted by the gatekeeper.

3.4. Results of the significance calculations

3.4.1. Initial result

Following equations (1) and (2), we first derive **s** = 3.5 (which is half of the 5 strings with inherent popularity 7 plus the 1 string with inherent popularity 10). And using the appropriate values, **m** = 13,430; **v** = 12,880; and **N** = 614,400; we derive **t** = 589,238. Thus:

$$P = s / t = 5.9 \text{ e-06 (preliminary)}.$$

3.4.2. Adjustment for spelling variation

There is an alternative Hebrew spelling for bin Laden, which omits the letter "yod", and is used about 50% of the time, according to a Hebrew Google search. Therefore, we halve our significance accordingly: This yields the final p-level:

$$P = 1.2 \text{ e-05, about 1 in 83,000.}$$

Figure 2: One example of a related matrix

4. Discussion

The current result should not be viewed in isolation. There are five other Torah code matrices on the same topic [6], involving a highly significant mixture of the collinear pattern (long phrases) and the two other patterns singled out for study in recent years: parallel and horizontal ELS's. Figure 2 is but one of these five. Because the large majority of the keywords found in these related matrices are a-priori (such as the most-cited words in Hebrew news accounts of 9/11), and because of a highly significant repetition of words or themes within the related matrices, some of them have estimated p-levels at least as significant as our case study.

The current method could be adapted to test a further implication of the Torah code hypothesis: if the codes are in fact an intentional communication rather than simply chance, then the capabilities of the codes' author far exceed those of human beings. It may be interesting for future studies of long phrases to examine perceived qualities of each phrase's author. For example, the reviewers could rate the "level of wisdom" of each phrase, as 'monkey', 'child', 'adult', 'prophet', or 'supernatural'. This rating actually would give a kind of combined intelligibility/wisdom assessment.

5. Conclusion

We have proposed a method for estimating the significance of a long ELS phrase string found in a text, based on its intelligibility as judged by a wide human review. Using this method, we have demonstrated that a particular ELS phrase string about bin Laden in the Torah has an estimated significance of 1.2e-05. This result, together with the related matrices mentioned in section 4, strongly suggest that this topic is intentionally coded in the Torah. In addition, because these results involve one of the most widely mentioned figures in today's news, this implicitly demonstrates that such significance levels are achieved quite readily in the Torah, without resorting to exhaustive searches. This adds compelling strength to the Torah code hypothesis.

6. Acknowledgements

7. References

[1] D. Witztum, E. Rips, Y. Rosenberg; Equidistant Letter Sequences in the Book of Genesis; *Statistical Science,* 9(3):429-438, 1994.

[2] H. Gans (2001); Torah Codes Primer; http://aish.com/seminars/discovery/Codes/codes.htm

[3] R. Haralick (2003); Testing the Torah Code Hypothesis; http://www.torahcodes.net/hypoth.html

[4] R. Haralick (2003); Torah Codes: Redundant Encoding; http://www.torahcodes.net/redun.html

[5] D. Witztum; Torah Codes; http://www.torahcodes.co.il

[6] A. Levitt (2004); Twin Tower Codes; http://www.torahcodes.net/alltwin.html