# Component Analysis of Torah Code Phrases

Art Levitt
Torah Code Research Group
Jerusalem, Israel
artlevitt23@yahoo.com

## Abstract

*In this paper, we develop a new tool, called Component Analysis ($CA$), to study the significance of long Torah Code phrases. $CA$ quantifies the relevance of such a phrase, by comparing its components (sub-phrases) to randomly constructed competitor phrases. In the process, we gain insight into how highly unusual it is to discover focused relevance among these randomly constructed competitors. Under the null hypothesis of no Torah Codes, we would therefore not expect to find focused relevance in the Torah Code phrases, but our experience suggests otherwise, as reflected in the highly significant example studied here. $CA$ lends itself well to being duplicated and verified by others, even those unfamiliar with Hebrew.*

## 1. Background

### 1.1. The Torah Code phrase of interest

The Torah Code phrase that we study here was discovered by Dr. Leib Schwartzman (see Figure 1). We will call the phrase $S_1$. Our goal is to estimate the significance of finding such a long and relevant Torah Code phrase about a single well-known topic (in this case, *bin Laden*).

### 1.2. Basic Torah Code definitions

Following the simple formula for all Torah Code building blocks, $S_1$ was formed by identifying equally spaced letters from the Torah (the first five books of the Hebrew Bible). This is called an ELS (equidistant letter sequence).

When finding an ELS in a text, we ignore all punctuation and inter-word spaces. For example, the ELS phrase "tin tops" can be found starting with the first "t" in the word "punctuation" in the preceding sentence, and using a *skip* of +4 (that is, counting forward every 4 letters).

### 1.3. Finding and displaying long phrases

Long ELS phrases like $S_1$ typically start with a given key word ("*anchor*"). We find the anchor's appearance as an ELS and attempt to extend it in both directions to form a longer ELS string, consisting of a meaningful phrase or phrases. The anchor of Figure 1, *bin Laden*, has a skip of 6598. The figure shows the underlying Torah text, of which the ELS is a part, with each row separated from the one above it by exactly 6598 letters. Near each word in the figure, we display the English translation, but all searching is done using the original Hebrew.

## 2. Description of the method

In this section we describe $CA$, and its language-independent variant, Relevance Analysis ($RA$). While relevance is a subjective judgement, we are able to quantify it by gathering the combined opinions of a large set of reviewers, using relative ranking and a large data set.

### 2.1. Component Analysis ($CA$): general description

$CA$ is a method of estimating the significance of an ELS phrase by comparing its component phrases, one at a time, to thousands of randomly constructed candidate competitors.

Only the anchor of our phrase is *a priori*, and the rest of the phrase is not. This requires that our method of gathering the competing phrases be as unrestricted as the method used to find the original phrase.

For our example phrase, $S_1$, we divide it into the following three components:

1. ארור (Cursed [is bin Laden])
2. ונקמה למשיח (and revenge [belongs] to the Messiah)
3. חרמה אכנך (I will dub you "Destruction")

0-7695-2521-0/06/$20.00 (c) 2006 IEEE


**Figure 1. A Torah Code phrase. The bin Laden "*anchor*" is extended by 22 letters**

Next, we determine the relevance of each component to the phrase's main theme (its anchor – *bin Laden* in our case). This relevance is estimated by comparing it to the relevance of the competitors.

The final step of $CA$ is to combine the results for all components, using the Fisher statistic [1]. A significant result would prompt rejection of the null hypothesis in favor of the alternative that long ELS phrases with focused relevance are readily found in the Torah.

These steps are described in more detail below.

## 2.2. Component Analysis: detailed steps

$CA$ compares each separate component of an ELS phrase, $S_i$, to many candidate competitors, as follows:

1. Divide $S_i$'s $w$ non-anchor words into $k$ components. A component can consist of from one to $n$ words (typically, 3 or less). Component boundaries are defined to coincide with the natural pauses of a phrase.

2. Compare each of $S_i$'s $k$ components to an appropriate list of candidate competitors (where the component itself is mixed in at an unmarked location, as are a sufficient set of control cases). For one-word components, the candidate list is derived from the 1640 major Hebrew roots listed in [7] (we choose an appropriate part of speech, and form the candidate from the root accordingly, so that the candidate fits the same context as the original component). For $n$-word components (n $\geq$ 2), the list of candidates is a set of $N$ $n$-word phrases, randomly extracted from a comparison population (defined below in 2.2.1).

3. Calculate a component relevance ratio for each of the $k$ components of $S_i$, which is simply the fraction of all candidates that are considered to be competitors of the component. A candidate is counted as a competitor of the component if it receives a relevance score that is higher than that of the component (see section2.2.3 for details). A candidate is counted as "0.5 competitors" if its relevance score is equal to that of the component.

4. Combine all $k$ component relevance ratios from (3) into an initial $p$-value, using the Fisher statistic [1]. This statistic gives the probability that the combination of the individual results could be as low as we observe.

5. Adjust by factoring in the skip of $S_i$, as follows. Typically, many ELSs exist for a given anchor in the Torah, but we favor those occurrences with lower skips. Given that $S_i$ was formed using the $j$th minimal occurrence of the anchor in the Torah, it requires a weighted Bonferroni adjustment detailed in the appendix, which is a function of $j$.

6. Adjust by factoring in the "difficulty of formation" ($DF$) of $S_i$. This is calculated by determining how often a randomly placed anchor can be extended to an ELS phrase of equal or longer length than $S_i$, with equal or greater average word length (with no requirement for relevance or even grammatical correctness - the extended string must simply consist of contiguous words found in the lexicon, described in 2.2.2 below). The $DF$ value is the number of such successful extensions of a random anchor, divided by the number of attempts. It is therefore independent of the relevance measures and is multiplied by the result from (5), to yield the final $p$-value for $S_i$.

### 2.2.1 The comparison population

We use two sources for obtaining candidate ELSs to compete with $S_i$'s multi-word components:

1. 60% of the candidates are extracted as ELSs from the segment of the Hebrew Bible immediately following

the Torah (this segment is truncated to have the same length as the Torah), and

2. 40% of the candidates are extracted as ELSs from the Torah itself. Since we are estimating the unlikelihood of finding relevant components situated precisely around an anchor in the Torah, it is logical to extract comparison phrases that also come from the Torah (but are situated around other anchor locations randomly chosen). This removes any uncertainties that may arise if the comparison phrases are extracted from other texts - perhaps those other texts have subtle structural differences from the Torah that would cause some side-effect differences in ELS behavior unrelated to the phenomenon under study.

   In addition, we do not enforce particular length restrictions on the candidates. Each is simply chosen so that the number of words that it contains matches the original component, but the average word length is permitted to vary.

We randomly identify and select $n$-word ELS phrases from each of these two sources, requiring only that every word be verified to be present in a lexicon of modern and ancient Hebrew, described next.

#### 2.2.2 The lexicon

The lexicon consists of all 40,000 unique words from the Hebrew Bible (excluding the book of Daniel, which contains many Aramaic words), and all unique words from all available issues of the online Hebrew news service, *Arutz-sheva*, from 2002 (almost 67,000 additional words).

#### 2.2.3 Relevance scoring

Our scoring protocol is very similar to that used in a previous study of the same phrase ([4]). We used a two-stage human review in that study to assess the intelligibility of $S_1$ among a large set of competitor phrases. This was accomplished with 91 reviewers, under double blind protocol. The current work has the following differences:

1. We obtained scores for each of the phrase's components separately.

2. We required a combination of intelligibility and relevance, not simply the former by itself.

3. We used less than half as many reviewers (for $CA$ and $RA$ combined), and stage 1 of our review was not double blind in this initial implementation. However, all review decisions and results can be independently validated by the reader at http://www.torahcodes.net/ca.html .

In our protocol, stage 1 is a pre-screen, which narrows the list of candidate competitors that are "promoted" to the stage 2 review. Stage 1 simply rejects unintelligible or irrelevant candidates.

Stage 2 reviewers rate each surviving candidate on a scale from 0 (irrelevant or unintelligible in their opinion) to 5 (profoundly relevant). For each candidate (and for the original component), its relevance score is defined as the number of reviewers assigning to it a score of 5.

We can now calculate the relevance ratio for the original component, which is simply the fraction of candidates with higher relevance scores, subject to the next section's correction.

#### 2.2.4 The "gatekeeper" correction

The stage one screening is similar to that done in the previous work, in that each candidate is subjected to a single reviewer, who acts as a kind of "gatekeeper". If this reviewer does not rate the candidate as viable, it does not pass the gate to stage 2, for further evaluation. In many cases this blocking is justified, but if we knew each candidate's "inherent popularity" (among a wide set of reviewers) we would observe some cases that were unjustifiably blocked by the gatekeeper's individual opinion. We use the simulation technique described in [4] to estimate the inherent popularities and thereby account for this gatekeeper effect.

### 2.3. Relevance Analysis ($RA$): general description

For non-Hebrew users, we define a language-independent variant of $CA$, called Relevance Analysis ($RA$). $RA$ is done completely in a language of one's choosing. Our initial $RA$ implementation uses the English translation for each component, and constructs the candidate competitors in English as well.

Just as we see with the Hebrew, a casual examination of the randomly constructed candidate competitors is instructive – it demonstrates how infrequently we observe truly competitive entries. The following entries are typical of the great majority, in their questionable degrees of intelligibility, and/or obvious non-relevance to the *bin Laden* anchor (they are candidate competitors for the component "*I will dub you destruction*"):

- You disabled their pollen
- He fed from nutcrackers
- We will bribe a spider

Following is a description of the individual steps of $RA$.

### 2.4. Relevance Analysis: detailed steps

$RA$ creates the components of $S_i$ in the same manner as $CA$, but it uses the translation into the language of choice.

$RA$ also uses similar methods to $CA$ to generate the lists of candidate competitors. For a single-word component of $S_i$, we again generate the candidates from the 1640 major Hebrew roots listed in [7], but translated into the language of choice.

For a multi-word component of $S_i$, we generate candidate competitors from a random sort of a dictionary of the language (rather than from random ELSs used in $CA$). Our initial implementation of $RA$ collects the first 2 nouns and the first 2 verbs from every page of the Oxford English Dictionary (2002), thereby extracting almost 2,000 words for each of these parts of speech. We consider these to be our "roots" and we randomly sort and combine them to form the candidate lists, via the following steps:

1. Our candidate list begins as a set of randomly combined words, with the $m$th word of each list member matching the part of speech of our component's $m$th word ($m$ = 1, ..., $n$). Each word is also optionally embellished in step (2).

2. Addition of grammatical context: a single word in Hebrew can indicate gender, tense, and number, and can also include pronouns, prepositions and conjunctions. In $S_1$, for example, the two-word component translated as "*I will dub you destruction*" contains only two roots - for *dub* and *destruction*. The prefix and suffix of the first root create the expanded context. $RA$ therefore randomly embellishes each word of a candidate to optionally include leading and/or trailing pronouns or other connectors at rates similar to that observed in random Hebrew ELSs. For example, the root "feed" may be embellished to become "I fed", or "he will feed" or it may be left unembellished. This may affect the intelligibility of the candidate, but it is approximately the same situation faced in $CA$ (by $S_i$, as well as by any Hebrew ELS that is used as a candidate). For the original component, interpretive aids, such as the word "[belongs]" are removed, so that only the basic structure competes: "*revenge is to Messiah*".

$RA$ compares each of $S_i$'s $k$ components to the candidate list in the same way as $CA$, using the same methods to calculate the results (steps (3) - (6) of section 2.2).

## 3. Connections to previous studies

The current techniques are built on the original work described in WRR [8]. Our study differs from WRR in (at least) two important matters:

1. We search the entire Torah, not only the book of Genesis.

2. We order the appearances of a keyword by the magnitudes of the observed skips.

The $CA$ and $RA$ methods reinforce our previous study ([4]), which obtained a $p$-value of $1.2 \times 10^{-5}$. We obtain greater significance in the current study (see following section) due partly to our focus on relevance. In addition, the current method ensures that proper credit is "accumulated", by considering each component's contribution to the overall rarity.

$CA$ and $RA$ should be applied only to those phrases that would be rated as *intelligible* by a significant portion of human subjects. We made that determination for $S_1$ in the previous work.

## 4. Applying $CA$ and $RA$ to the bin Laden phrase

The following sections present the $CA$ and $RA$ results for each component and for the combined outcomes.

### 4.1. The $CA$ result

The $CA$ procedure yielded the following relevance ratios for each of $S_1$'s components (with the numerators already increased via application of the gatekeeper correction):

ארור (Cursed): 0.5/1640 = $3.0 \times 10^{-4}$

ונקמה למשיח (and revenge [belongs] to the Messiah): 119/12500 = $9.5 \times 10^{-3}$

חרמה אכנך (I will dub you "Destruction"): 30.5/12500 = $2.4 \times 10^{-3}$

Combining these 3 results with the Fisher statistic, we obtain an initial unadjusted $p$-value of $1.4 \times 10^{-6}$

We now apply the two adjustments detailed in section 2.2. The $S_1$ anchor is the fifth minimal occurrence of this ELS in the Torah, which requires an adjustment factor of 39.6. The *difficulty of formation* ($DF$) factor is a conservative $7.2 \times 10^{-3}$. This factor was calibrated by using a flexible lexicon, which artificially permitted formation even for a long phrase that contains a rare word - one that is not present on the original lexicon but is present on an alternative, 50% inflated version.

The final $p$-value after these two adjustments is $4.0 \times 10^{-7}$, or 1 in 2.5 million, the probability that such a long ELS phrase could be found surrounding any given bin Laden ELS anchor, in a Hebrew text the size of the Torah, and that this phrase would be intelligible, and consist of components that are together as relevant to bin Laden as $S_1$, merely by chance.

### 4.2. The $RA$ back-up result

Using English to evaluate the concepts conveyed by the Hebrew, $RA$ yields the following relevance ratios for each of $S_1$'s components (with the gatekeeper correction already applied to the numerators):

*cursed*: 71.5/1640 = $4.3 \times 10^{-2}$

*revenge is to the Messiah*: 1.5/8000 = $1.9 \times 10^{-4}$

*I will dub you destruction*: 21/8000 = $2.6 \times 10^{-3}$

After combining using the Fisher statistic and applying the two adjustments we obtain a final $p$-value estimate of $1.1 \times 10^{-6}$, about 1 in 0.9 million. This is comparable to the $CA$ result obtained above. In both cases, the null hypothesis of no Torah Code effect is clearly rejected.

We believe that the use of $RA$ in other languages would yield similar results.

## 5. Beyond the $p$-values

The low $p$-values only partly reflect the high relevance of the words that comprise our ELS phrase. We observe just how precise these words are, when we see their similarity to language used in the Bible itself. One example is the use of *dubbing* or *nicknaming* (Isaiah 45:4). We see further precision with the Biblical use of our specific synonym for *destruction*: it is actually used as a name to dub something - in this case the destroyed cities in Numbers 21:3 and Judges 1:17. In addition, our ELS seems to echo the language in Deuteronomy 32:35 ("*vengeance [belongs] to Me*").

We do not presume to have the ultimate interpretation of meanings, but rather evidence of the existence and rarity of relevance in this example ELS phrase.

## 6. Conclusion

$CA$'s strength is in its simplicity, and its ability to mutually reinforce results obtained from previous work [4]. The simplicity derives from the fact that the analysis is limited to a single long phrase - a single ELS. The reinforcement extends to other more complex phenomena such as clustering of many ELSs related to the Twin Towers attack ( [5], [3]).

## Appendix

We apply the weighted Bonferroni inequality described below to situations where we have an ordered series of trials.

Consider testing (null) hypothesis $H_1, H_2, \ldots$ with corresponding $p$-values $P_1, P_2, \ldots$. Let $w_1, w_2, \ldots$ be a sequence of weights with $w$ their sum. If a specific hypothesis $H_i$ is rejected when $P_i \leq \alpha \cdot w_i/w$, then the weighted Bonferroni inequality

$$\Pr\{\cup(P_i \leq \alpha \cdot w_i/w)\} \leq \alpha$$

ensures that the probability of rejecting at least one hypothesis when all are true is no greater than $\alpha$.

We cannot choose $w_i = 1/i$ because the harmonic series is diverging. Therefore we set $w_i = 1/[(i+1)\cdot(\ln(i+1))^2]$ to obtain a convergent series.

Unlike the usual form of the weighted Bonferroni inequality ( [6], [2]) this form assigns to the null hypothesis $H_i$ a weight which is dependent only on $i$ and not on the total number of null hypotheses. Therefore, it can be used in a situation where the number of hypotheses is not pre-specified, but their order is established. We meet this situation when investigating the ELS strings generated by consecutive appearances of a given anchor.

## Acknowledgments

We wish to thank Rabbi Earl David and Yuri Pikover for their generous support; and Harold Gans, Professor Robert Haralick and Professor Eliyahu Rips for their valuable technical advice.

## References

[1] R. Elston. On Fisher's method of combining $p$-levels. *Biometrical Journal*, 33:339–345, 1991.

[2] Y. Hochberg. A sharper Bonferroni procedure for multiple tests of significance. *Biometrica*, 75:800–802, 1988.

[3] A. Levitt. Torah Codes primer and survey of the latest research. http://www.torahcodes.net.

[4] A. Levitt, N. Bombach, H. Gans, R. M. Haralick, L. Schwartzman, and C. Stal. Long phrases in Torah Codes. In *2nd Annual Speech and Language Conference*, U. of Belgrade, 2004. http://www.torahcodes.net/papers.html.

[5] E. Rips and A. Levitt. The twin towers cluster in Torah Codes. In *Proceedings of the 18th International Conference on Pattern Recognition*, August 2006.

[6] R. J. Simes. An improved Bonferroni procedure for multiple tests of significance. *Biometrica*, 73:751–754, 1986.

[7] A. Solomonick and D. Morrison. *Maskilon I: Hebrew English Dictionary based on Verb Roots*. Gefen, Jerusalem, 2001.

[8] D. Witztum, E. Rips, and Y. Rosenberg. Equidistant letter sequences in the book of Genesis. *Statistical Science*, 9(3):429–438, August 1994.