

## An Overview of Syntax Features in the Libronix DLS

---

For the past year, we at Logos Bible Software have been working to add support for syntax to the Libronix Digital Library System (LDLS) publishing platform. Where a morphologically-tagged text is useful for studying word-level features, syntax markup and searching takes linguistic study beyond words to the level of phrases, and clauses. This higher-level markup allows visualization of the structure of phrases and clauses within a text; it makes possible searches that cannot be performed with word-level morphology; and by identifying and labeling all the parts of a clause, it makes the job of exegesis just a little bit easier.

### What is Syntax?

---

By SYNTAX we generally mean any linguistic structure that is higher than the word, up to (and perhaps including) discourse units. If grammatical structure is seen on a continuum, with words, which have zero internal grammatical structure, at one end and discourse units, which have maximum internal grammatical structure at the other, then syntax occupies the intermediary levels of the continuum: phrases, clauses, speech/dialogic/quotative frames, and sentences.<sup>1</sup> Whenever two or more words are gathered together, syntax is there.

Morphologically-tagged texts can be pressed into service to find some syntactic features, by searching for the co-occurrence of multiple morphological contours in close proximity. Furthermore, most electronic morphological databases of the Bible on the market today are really morphosyntactic because they mark not only morphology but some rudimentary syntax categories as well. However, this method is imprecise for two reasons: 1) The boundaries of the syntactic structures, such as the extent of prepositional phrases or clauses, are not marked; 2) the syntactic relationships between the structures, such as when a noun functions as the subject of a clause versus when it functions as the object of a prepositional phrase, is not marked. Morphological databases are a step in the right direction, but they are insufficient for studying syntax. By their very design, they keep study at the level of individual words.

### Syntax Presentation in LDLS

---

The upcoming 3.0 version of the Libronix Digital Library System technology will have support for syntax resources that are fully integrated with the rest of a user's library. Unlike most resources in the Libronix DLS, which are single, self-contained units, each syntax resource consists of several parts: 1) a RUNNING TEXT resource, which is equivalent to a morphologically-tagged text, and which holds all of the words of the text in sequence; 2) some number of VISUALIZATIONS of the syntactic information; and 3) an INDEX of syntactic structures used for searching.<sup>2</sup>

---

<sup>1</sup> Or whatever terminology is specified by a given syntax database. Libronix does not impose a fixed vocabulary of syntax constructions.

<sup>2</sup> Additional resources, such as linguistic commentaries, apparatuses, or syntax notes can also refer to or be part of the collection of resources that comprise a syntax database. However, these are periphery. The running text, visualizations, and search index are the nucleus.

## Running Text

A running text resource that is part of a syntax database is nothing more than a simple Libronix resource, a scrolling book that begins at the beginning and runs until it reaches the end. However, each word-level segment of syntactic analysis is identified with a unique address, called a WORD NUMBER, which places that that word in sequence in its natural location within the running text. Using this word number, the various parts of the syntax database can keep track of the location of each word in the database. When, for example, a syntactic search is performed on the syntax index, the running text is highlighted.

Furthermore, the word number allows multiple running text resources to be associated with a syntactic database, so that any equivalent editions of the annotated text can be considered part of the group.

Usually, there is one running text resource per syntax database that contains the words and the word-level morphosyntactic markup for the database. Thus, the running text is treated like any other in the Libronix DLS: It can be searched with the same morphological or graphical search interfaces, can be hyperlinked to and from other resources, and so on. It is a book like any other.

## Visualizations

While a running text is just a book like any other, a SYNTAX VISUALIZATION resource is a diagram of some or all of the syntactic information in the database. It represents a way to visualize, or see, the syntax markup. Any number of visualizations can be part of a single syntax database, and each visualization of the data is cataloged in the DLS with its own metadata (title, author, *etc*) and accessed through the My Library dialog like any other resource.

In theory, any number of visualization methods or presentation formats could be linked with a single syntax markup database. In the initial release, however, there is one graphical visualization resource driver: The DIRECTED ACYCLIC GRAPH resource. In the following sections, I will describe why that particular method of visualization was chosen over several others: Reed-Kellogg (“traditional”) diagrams, sentence flow diagrams, and syntax trees.

### Reed-Kellogg Diagrams

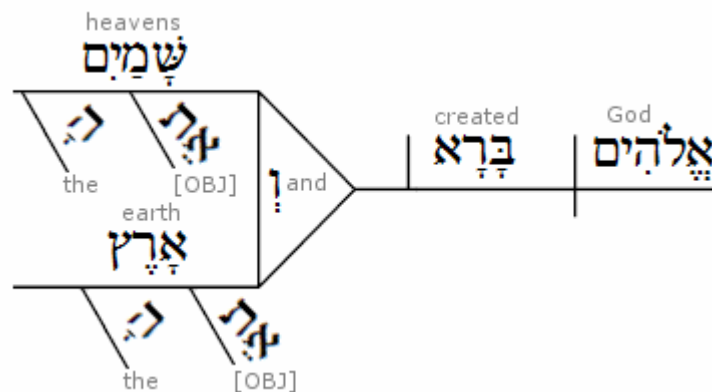


FIGURE 1. A Reed-Kellogg diagram of a clause from Genesis 1:1

Also called “traditional” sentence diagrams, Reed-Kellogg diagrams have been in widespread use, especially in America. The diagramming system now known as the Reed-Kellogg system was first copyrighted by Alonzo Reed and O. H. Hall in 1868; a few years later Reed and Brainerd Kellogg popularized this diagramming scheme by including it in their elementary English grammar, *An Elementary English Grammar*. Since then, they have seen widespread pedagogic use for teaching English grammar at primary, secondary, and beyond.

These diagrams have much to commend them. They are familiar to many people, especially those whose only exposure to syntax is limited to grade-school grammar classes or seminary Greek courses. They have withstood the test of time; no system of annotation retains currency under widespread use for more than one hundred years without being generally useful. Perhaps most importantly of all, the rational nature of these diagrams, which cause the student of a language to ponder and pore over the “logical” structure of every clause that is diagrammed, are useful for pedagogy. Reed and Kellogg say as much in their preface to their grammar:

We invite attention to our system of Diagrams. . . . They show, at a glance, the relation of every word and division of the sentence, and are a device attractive and helpful to the pupil in the preparation of his written lessons. They mechanical work interests him, and lures him on to abstract analyses which he could not make without such aid.

The fact that the pictorial diagram groups the parts of a sentence according to their offices and relations, and not in the order of speech, has been spoken of as a fault. It is, on the contrary, a merit, for it teaches the pupil to look through the literary order and discover the logical order. He thus learns what the literary order really is, and sees that this may be varied indefinitely, so long as the logical relations are kept clear. . . . The diagram drives the pupil to a most searching examination of the sentence, brings him face to face with every difficulty, and compels a decision on every point.<sup>3</sup>

However, this same utility in teaching is the primary drawback of this system for the static representation of syntactic information. Whether or not making sentence diagrams indeed “interests” the student or “lures him on” to deeper analyses, it is certainly the case that the primary teaching benefit of such a diagram accrues from the *exercise* of making the diagram. They are a help for the “preparation of . . . written lessons.” They do not, as a result, make a very good reference.

Since this diagramming system was developed for teaching English grammar, it is useful for teaching English grammar. It has also been well-used to teach Latin and Greek, insofar as the classical languages are similar in construction to English — and insofar as various prescriptive grammarians have attempted to conform English grammar to the rules of either Latin or Greek! However, this model is not as well-suited to marking other languages as it is to English. The system can be adapted, by adding new hooks and angles, but in the end this amounts to little more than adding epicycles to the Ptolemaic model of the cosmos.

Finally, the plethora of shapes and rules for applying them that have grown into the system over time makes them difficult to encode and to draw programmatically. It is possible, of course, but difficult. Some more abstract system of representation is indicated.

---

<sup>3</sup> Reed, Alonzo and Kellogg, Brainerd. *An Elementary English Grammar: Revised Edition*. 1896. New York: Effingham Maynard & Co. pp 5–6.

The Libronix DLS has good support for making Reed-Kellogg diagrams in the Sentence Diagrammer addin. After all, this is the original intent of the system's inventors, that students be encouraged in the *exercise* of diagramming in order to learn grammar in general and to understand the inner workings of a particular sentence. Still, it is not the method that we chose to represent pre-modeled syntax markup.

### **Phrase-Structure Diagrams/Flow Diagrams/Outlines**

It is possible to indicate the “structure” or “flow” of a given text by adding line breaks and indentions to show the relationships between the various parts of the text. This method is popular in hermeneutics and homiletics, because it quickly and accessibly highlights the structure of a text: the flow of an argument, the general contour of a narrative, the levels of a chiasm, the stanzas and stiches of a poem. Such presentation of text structure as a sort of indented outline is easily apprehended, as well. Last but not least, flow diagrams keep the text in its natural order, which is useful for reading and understanding as much as analysis, rather than chopping it up into its various constituents and rearranging it “logically.”

From a programmatic standpoint, the structure is certainly abstract: All that must be tracked or drawn is the position of the line breaks and the level of indentation of each line. In fact, this sort of analysis can be easily performed in a word-processing document, and that level of presentational formatting is already supported by Libronix DLS resources.

However, while the simplicity of an indentation-based scheme is certainly a strength, it is also a fatal weakness. Since these sorts of diagrams generally do not provide labels for the various parts of the sentence or clause, they are not sufficient for syntactic markup. Furthermore, they tend not to break the text down to a very fine level; if they did, nearly every word would be on its own line. A thorough syntactic analysis of a given text will require more information. It will need to label every part and show the relationships.

A syntax diagram must be able to show how smaller structures form together into larger structures and how those structures are interrelated.

### **Syntactic Trees**

Linguists generally use syntax trees to mark syntactic information. This is a very abstract and flexible way of presenting information, beloved by mathematicians, linguists, and computer programmers alike. A TREE is “a mathematical structure that can be viewed as either a graph or a data structure. . . . A tree graph is a set of straight line segments connected at their ends containing no closed loops (cycles).”<sup>4</sup>

Regardless of one's views on the extent that modern linguistic theory should or should not inform biblical studies, it must be admitted that the tree diagram is a flexible and powerful way to label all the parts of a thorough syntactic analysis. Reed-Kellogg diagrams require the student to master a large vocabulary of shapes that stand in for syntactic labels; syntax trees, on the other hand, contain only points and lines, and the points are usually given a descriptive label.

---

<sup>4</sup> “Tree — From MathWorld.” <http://mathworld.wolfram.com/Tree.html>

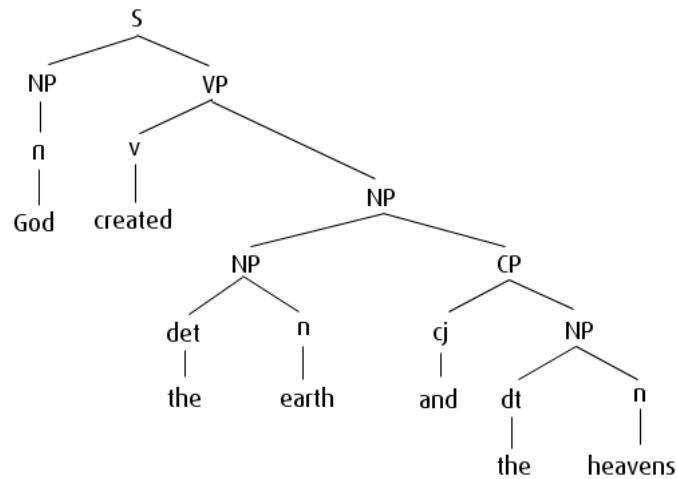


FIGURE 2. A syntax tree diagram of a clause in Genesis 1:1

In a syntax tree, every syntactic structure is given a label, and the label points to its constituent parts. In FIG 2, above, the sentence (S) consists of a noun phrase (NP) and a verb phrase (VP). These labels, again, are arbitrary; another system of markup could assign different categories altogether. In FIG 3, below, a definite noun of geography (**n:geog/def**) points to the two words that form its constituents: *the* and *land*.

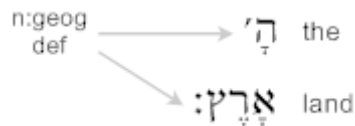


FIGURE 3. A labeled structure points to its constituents.

**Syntax trees are programmable.** A great variety of syntax trees and methods for generating them exist. The particulars of any given system are outside the scope of this paper. Suffice it to say that because of their clear and rigidly-defined mathematical properties, trees are a natural fit with any electronic syntax solution. From any point on a tree, it is a trivial mathematical operation to retrieve all of its constituent parts (its children) and all of the structures that it is a constituent of (its parents); this is a great aid in adding interactive features, such as contextual highlighting (see below).

**Syntax trees are flexible.** Since the labels are arbitrary, syntax trees are capable of visualizing virtually any set of markup categories applied to virtually any text. They are thus not only language-independent, but markup-independent as well.

**They can't do everything.** Unfortunately, human language is sometimes not logical and well-behaved enough to fit neatly within the confines of a rigidly defined mathematical structure. Since mathematically, trees are a subset of graphs,<sup>5</sup> a graph-based visualization fully supports trees as well. Particularly, the *Andersen-Forbes Phrase Marker Analysis of the Hebrew Bible*<sup>6</sup> markup cannot be fully expressed as simple mathematical trees.

<sup>5</sup> That is, all trees are graphs, but not all graphs are trees.

<sup>6</sup> Andersen, Francis I. and Forbes, A. Dean. Upcoming publication.

## Directed Acyclic Graphs

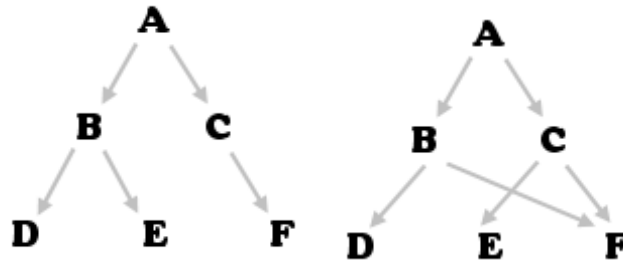


FIGURE 4. Both are DAGs, but only the one on the left is a tree.

A DIRECTED ACYCLIC GRAPH (DAG) is a diagram or structure that consists of points (or labels) and arrows that contains no cycles, which is another way of saying that one chooses a label and follows the arrows in the direction that they are pointing, the same label on the graph will never be visited more than once. A DAG differs from a tree in one key respect: Each label is allowed to have multiple arrows pointing into it, where in trees, each label is allowed only one arrow pointing in (see FIG 4, above).

DAGs are capable of displaying complex relationships without rearranging the natural order of the text.<sup>7</sup>

**Tangling.** Sometimes, words that belong to the same grammatical or syntactic structure are broken up and cast to the far ends of a sentence or clause. In the sentence, “I would like *to* quickly, if at all possible, *address* the issue of split infinitives,” the infinitive *to address* is split. When diagramming such a structure, three possibilities exist: 1) Rearrange the text so that the parts of each structure are adjacent; 2) Keep the text in its original order and use multiple labels to label each part of the structure (eg, A, B etc); 3) Keep the text in its original order and use a single label to mark the structure. For Andersen-Forbes, we must use option 3:

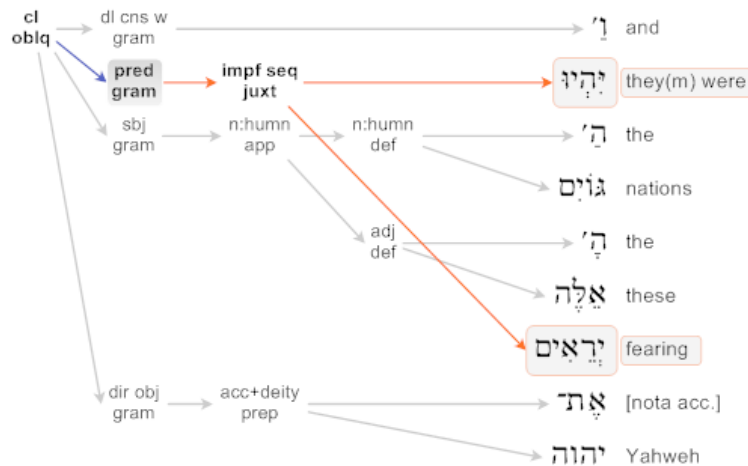


FIGURE 5. A split construction.

<sup>7</sup> For a more detailed explanation, see Andersen, F. I. and A. D. Forbes, "Hebrew Grammar Visualised: I. Syntax," *Ancient Near Eastern Studies* XL (2003): 43-61.

**Multidominance.** Sometimes one grammatical structure modifies several others. For example, in Genesis 1:5, *It was evening and it was morning, day one*, the noun phrase *day one* modifies not only the first clause (*it was evening*) but also the second (*it was morning*). In effect, *day one* belongs to both clauses, because it modifies both *evening* and *morning*.

This sort of multiple modification can be notated in a DAG by MULTIDOMINANCE, that is, by letting one construction be part of multiple other constructions. In the DAG in FIG 6 below, the noun of time *day one* is dominated by (owned by, part of) two clauses in the verse, indicated by the two arrows leading into the label:

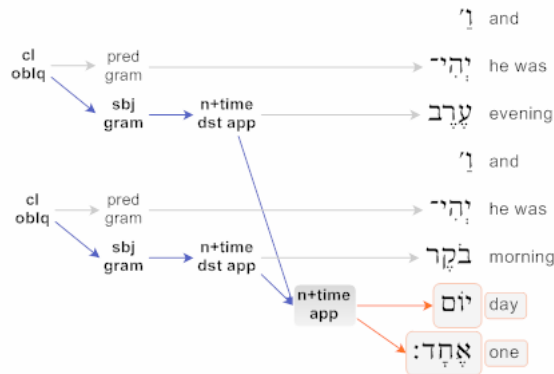


FIGURE 6. *Multiple modification marked by multidominance.*

Certain kinds of ellipsis can also be marked by multidominance in DAGs. In Obadiah 19, we encounter two clauses: “Those of the Negeb shall possess Mount Esau, and those of the Shephelah shall possess the land of the Philistines.” (Ob 19, ESV) In the Hebrew, however, the verb is not repeated in the second clause. In effect, one verb is shared between two clauses, named in the first, not named in the second.

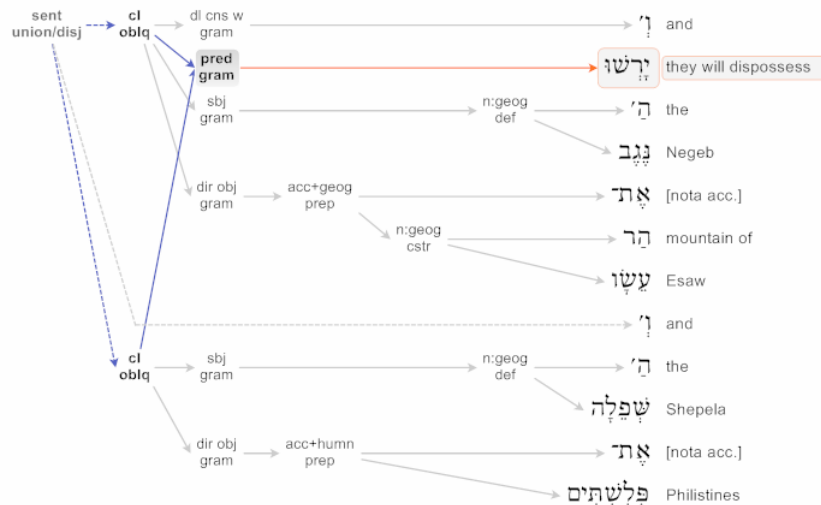


FIGURE 7. *Ellipses marked by multidominance.*

## Index & Searching

The third component of a Libronix DLS syntax project is the index. This index is implemented as a specialized database, and unlike the running text and visualization resources, is never shown to users. The index database is used for searching, not for display.

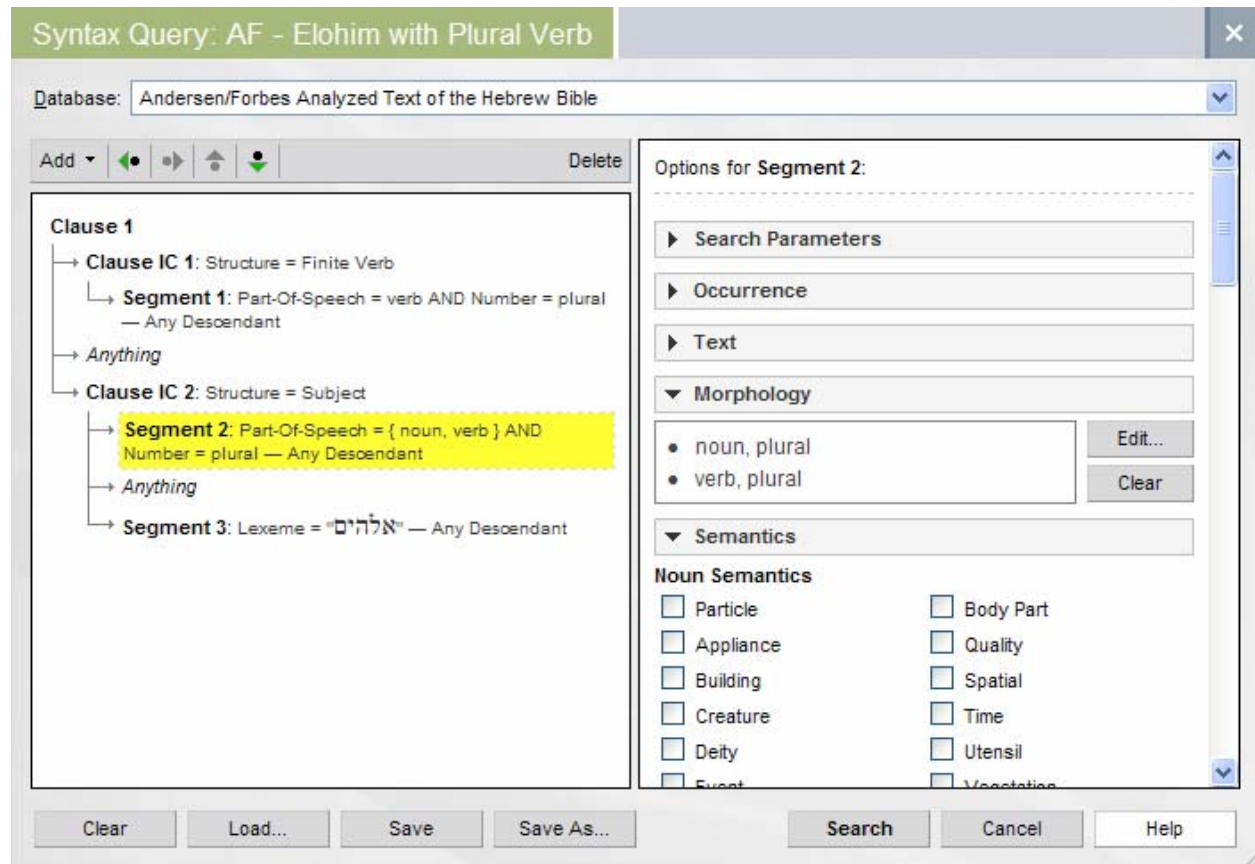


FIGURE 8. *The Libronix DLS Syntax Search dialog.*

The Syntax Search dialog consists of two panels. On the left, the user creates a tree-like diagram that represents the search query he wishes to find. In FIG 8, above, the query general query is, “Find all clauses with a plural verb predicator where *elohim* is the subject.” The word *elohim* (“God”) is always plural in form, but is usually grammatically and contextually singular. The query in FIG 8 will find all clauses where the word *elohim* is used with a plural verb, where it should possibly be translated “gods” and not “God.”<sup>8</sup>

The query in FIG 8 reads as follows: Find all clauses that contain: 1) a clause-immediate constituent that is a “Finite Verb” that contains a text segment that is a plural verb; 2) zero or more possible intervening constructions; 3) a clause-immediate constituent that is a “Subject” within the clause that does *not* contain any plural nouns,<sup>9</sup> but does contain the word *elohim*.

<sup>8</sup> Each instance would need to be carefully evaluated in context, of course. This is a grammatical rule, not a theological one!

<sup>9</sup> This constraint removes several false hits where the subject of the clause contains the word *elohim* but is in fact plural, such as *beney ha-elohim*, “sons of God”.



The right-hand panel contains specific options for each term in the query panel on the left. In FIG 8, above, the term **Segment 2** is selected, and its options appear on the right. The options for each kind of term are arbitrary; clauses need not have the same options (or user interface to set those options) as words.

Results of the query are given in a separate window, with the full text of any verses the query intersects given, with the full extent of the query highlighted in blue, along with any terms that were marked in the query to receive special highlighting colored in red, as in FIGURE 9, below.

Syntax Search Results 2

ExecuteQuery: 4656/1 GetPreviews: 1687/1

ESV

13 And when God caused me to wander from my father's house, I said to her, 'This is the kindness you must do me: at every place to which we come, say of me, He is my brother.'

Gen 20:13

וַיְהִי כַּאֲשֶׁר הִתְעוּ אֹתִי אֱלֹהִים מִבֵּית אָבִי וְאָמַר לָהּ זֶה חֲסִדְךָ אֲשֶׁר תַּעֲשִׂי עִמָּדִי אֵל כָּל־הַמְּקוֹם אֲשֶׁר נָבֹא שָׁמָּה אֲמַר־לִי אָחִי הוּא:

7 and there he built an altar and called the place El-bethel, because there God had revealed himself to him when he fled from his brother.

Gen 35:7

וַיִּבֶן שָׁם מִזְבֵּחַ וַיִּקְרָא לְמָקוֹם אֵל בֵּית־אֵל כִּי שָׁם נִגְלוּ אֵלָיו הָאֱלֹהִים בְּבָרְחוֹ מִפְּנֵי אָחִיו:

9 For every breach of trust, whether it is for an ox, for a donkey, for a sheep, for a cloak, or for any kind of lost thing, of which one says, 'This is it,' the case of both parties shall come before God. The one whom God condemns shall pay double to his neighbor.

Exod 22:8

עַל־כָּל־דְּבַר־פְּשָׁע עַל־שׂוֹר עַל־חֲמוֹר עַל־שֶׂה עַל־שְׁלָמָה עַל־כָּל־אֲבָדָה אֲשֶׁר יֹאמַר כִּי־הוּא זֶה עַד הָאֱלֹהִים יָבֹא דְבַר־שְׁנֵיהֶם אֲשֶׁר יִרְשִׁיעַ אֱלֹהִים יִשְׁלַם שְׁנַיִם לְרֵעֵהוּ:

26 Saul also went to his home at Gibeah, and with him went men of valor whose hearts God had touched.

1 Sam 10:26

וְגַם־שָׂאוּל הִלְךָ לְבֵיתוֹ גִּבְעָתָה וַיִּלְכוּ עִמּוֹ הַחַיִל אֲשֶׁר־נָגַע אֱלֹהִים בְּלִבָּם:

23 And who is like your people Israel, the one nation on earth whom God went to redeem to be his people, making himself a name and doing for them great and awesome things by driving out before your people, whom you redeemed for yourself from Egypt, a nation and its gods?

2 Sam 7:23

וּמִי כַעֲמֹךָ כִּי־שָׂרָא לְגוֹי אֶחָד בְּאֶרֶץ אֲשֶׁר הִלְכוּ־אֱלֹהִים לְפָדוֹת־לוֹ לְעַם וְלָשׂוּם לוֹ שֵׁם וְלַעֲשׂוֹת לָכֶם הַגְּדוּלָה וְנִרְאוֹת לְאַרְצֶךָ מִפְּנֵי עַמְּךָ אֲשֶׁר פָּדִיתָ לְךָ מִמִּצְרַיִם גּוֹיִם וְאֱלֹהֵיוֹ:

2 Then Jezebel sent a messenger to Elijah, saying, "So may the gods do to me and more also, if I do not make your life as the life of one of them by this time tomorrow."

1 Kings 19:2

וַתִּשְׁלַח אִיזָבֵל מֵלֶאֱדָה אֶל־אֵלִיָּהוּ לֵאמֹר כֹּה־יַעֲשׂוּן אֱלֹהִים וְכֹה יוֹסִפוֹן כִּי־כַעַת מִחַר אֲשִׁים אֶת־נַפְשְׁךָ כְּנַפְשׁ אֶחָד מֵהֶם:

10 Ben-hadad sent to him and said, "The gods do so to me and more also, if the dust of Samaria shall suffice for handfuls for all the people who follow me."

1 Kings 20:10

וַיִּשְׁלַח אֵלָיו בֶּן־הַדָּד וַיֹּאמֶר כֹּה־יַעֲשׂוּן לִי אֱלֹהִים וְכֹה יוֹסִפוּ אִם־יִשְׁפַק עָפָר שְׂמֵרוֹן לְשַׁעֲלִים לְכָל־הָעָם אֲשֶׁר בְּרַגְלֵי:

FIGURE 9. Syntax search results.