

Can I Trust My Bible Software? Why Bible Searches May Not Find What You Expect

By Dr. Harry Alan Hahne
Professor of New Testament – Gateway Seminary

Evangelical Theological Society Far West Regional Conference. Ontario, California. April 21, 2017

Bible software enables more rapid and in-depth research than is practical with printed reference books. Yet computer-assisted Bible study has several potential sources of error in addition to the common errors of traditional research methods. If Bible software is not used carefully, you may simply get the wrong answer more quickly! As a result, pastors can mislead their congregations and scholars can be discredited by publishing inaccurate research.

Sometimes the same search produces varying results with different Bible programs or with different digital versions of the biblical text searched with the same software. Bible programs differ in their search capabilities and assumptions. Searches in the original languages of the Bible open up additional potential sources of error, because various digital Greek and Hebrew biblical texts follow differing morphological and syntactical tagging philosophies.

A user who is unaware of these characteristics of the software and texts may assume that a search is correct, when in fact it has hidden errors. Since the encoding of the biblical text and the design of the search software may be poorly documented, trial and error is often necessary to determine how the software works. Online forums are full of discussions about why searches did not return the results that the user expected.

This paper explores several reasons searching Bible software can produce incorrect results: (1) characteristics of the digital biblical texts; (2) capabilities of the search software; and, (3) common errors in using the software. Some suggestions on how to use your software effectively will help you rely on your research.

This paper focuses on searching the Greek New Testament (NT), with an emphasis on searches for grammatical constructions. Logos Bible Software version 7 is considered in depth, including the Morphological Query Builder, which is currently only available through the Logos Now subscription service.¹ There are also some comparisons to other Bible software.² Nevertheless, the principles are applicable to any Bible software as well as searches in translations or in the Hebrew Bible.

Characteristics of Digital Biblical Texts

Search results will clearly depend on the biblical text being searched. Yet there are also several less obvious characteristics of digital biblical texts that can affect the accuracy of searches.

¹ This is abbreviated “Morph Query Builder” in the Logos Help document and throughout this paper. This powerful search feature became available through the Logos Now subscription service starting with Logos 7.2.

² Logos 7.6 was used for testing. Other programs that will be mentioned are Gramcord for Window 2.4cx, BibleWorks 9, Accordance 10, and Biblio 8 (formerly called Bible Windows).

1. The Bible Version

Numerous digital versions of the Greek NT are available. Logos Bible software alone offers dozens of Greek NT editions.³ These texts vary in two primary ways: (1) the surface text of the Greek NT; and (2) the tagging of linguistic, morphological, and syntactical features.

Variations between Greek texts are due to the fact that ancient books were hand-copied prior to the invention of the printing press. Over the last several hundred years, Bible scholars have published several editions of the Greek NT that represent decisions about which of the minor variations between the thousands of ancient manuscripts most likely represents the original wording. As new manuscripts are discovered and scholars do further analysis of the manuscript variations, new scholarly editions of the Greek NT are published.

Most scholars base their research on the Nestle-Aland (NA) or the United Bible Societies (UBS) Greek texts. Some occasionally use the Society of Biblical Literature Greek New Testament (SBLGNT), which is only slightly different. These editions represent the result of a comparison of thousands of ancient NT manuscripts. Few modern scholars use the Textus Receptus, which was based on a few late medieval manuscripts.⁴

It is important to know which Greek text is used by your Bible program. Since the UBS 4th edition (UBS4), NA 26th edition (NA26) and NA 27th edition (NA27) all share the same Greek text, search results on these texts should be the same.⁵ The Greek text of the more recent UBS5 and NA28 has only minor differences from these earlier editions in 34 verses in the Catholic (General) Epistles.⁶ The SBLGNT has about 540 variations from the NA Greek text, but it is much closer to this edition than it is to the Textus Receptus.⁷ The Textus Receptus

³ Some of the texts available for Logos Bible Software are listed at <http://www.logos.com/greek/nt/texts/>. My own Logos collection has 31 Greek New Testament texts, not including reverse interlinear English Bibles.

⁴ Erasmus produced the first edition of his Textus Receptus Greek New Testament in 1516 based on only six Greek manuscripts. All manuscripts came from the 12th century or later and all except one represented the Byzantine text type (also called the Majority Text). By contrast, more modern editions such as the NA, UBS and SBLGNT are based on nearly 6000 extant Greek manuscripts, including fragments from the early 2nd century and complete texts from the early 4th century. The King James Version is a translation of the Textus Receptus, while most recent NT translations use NA or make editorial decisions based on comparing the richer collection of earlier manuscripts that are available today.

⁵ The NA28 and UBS5 Greek New Testaments share the same Greek text, apart from minor differences in punctuation, paragraphing and capitalization. Similarly, the NA26, NA27, UBS3, and UBS4 Greek New Testaments share the same Greek text. The main difference between these editions is that the “critical apparatus” in the print editions differ in the way they list the variations between the thousands of ancient manuscripts. These differences do not affect the searchable text in Bible software. For a discussion of the differences between these editions, see <https://www.academic-bible.com/en/home/scholarly-editions/greek-new-testament/comparison-na28-ubs5/> and <http://www.logos.com/support/lbs/na-v-ubs/>.

⁶ For a list of differences between the NA28 and the NA27, see <http://bibleworks.oldinthenew.org/?p=1133>. The German Bible Society Web site (<http://www.nestle-aland.com/en/the-28-edition/revision-of-the-catholic-letters/>) also has a less easy to read but more informative list of the differences.

⁷ There are 5959 differences between the SBLGNT and the Robinson-Pierpont Byzantine text (<http://sblgnt.com/about/introduction/>). Although the Textus Receptus is not identical to the Byzantine text type (Majority Text), this is its primary source. By contrast, there are only about 540 variations between the SBLGNT and the NA (<https://www.logos.com/product/8486/the-greek-new-testament-sbl-edition/>).

differs in thousands of places from these more recent editions and thus could have substantially different search results.

2. Morphological Tagging Systems

A *tagged text* attaches descriptive information (*tags*) to every word in the text. Bible software can search on the tags as well as the *surface text* of the Bible. A digital text may have multiple tags containing various types of information about words, phrases and clauses. A few digital Greek NT texts also classify the syntactical relationships of words, phrases and clauses. However, this discussion will focus on the tagging of the morphology of individual Greek words.

Greek words change their spelling (or *morphology*) to indicate the grammatical function of words in the sentence. Greek morphological tagging systems range on a continuum from primarily *formal* classifications (purely morphological) to more *functional* classifications (syntactical). The more a tagging system tends toward the functional end of the spectrum, the more subjective the classifications become. Thus, even if two digital texts use the same Bible version, search results may vary depending on the tagging system used.

There are several Greek morphological tagging systems commonly used in Bible software:

Morphological Tagging	Characteristics of Tagging System
Logos Greek Morphology ⁸	Largely formal. Conjunctions, particles, and adverbs have functional subtypes that are determined from context.
Gramcord Morphology ⁹	Largely formal. Alternatives are included for words with debatable morphology.
Friberg Morphology, used in the Analytic Greek New Testament (AGNT) ¹⁰	Frequently functional, based on discourse analysis. Several unusual functional categories are used.
Swanson Greek Morphology	Mixed. Largely formal, but some functional categories are based on “what the form ‘is being used as’” in the sentence. ¹¹ Formal parsings are often included as an alternative tag.

⁸ Logos Greek Morphology is used by numerous Logos Greek New Testament resources with essentially the same text, including NA27, NA28, UBS4, UBS5, Lexham Syntactic Greek New Testament, Lexham Clausal Outlines, and Opentext.org (since Logos 4). It is also used by SBL Greek New Testament, Westcott and Hort, Scrivener’s Textus Receptus, and Stephen’s Textus Receptus, all of which use a different Greek text than the NA/UBS.

⁹ The NA27 with Gramcord Morphology is available for Gramcord Bible Software, Accordance and Logos. The Gramcord software manual explains the tagging philosophy and specifies how ambiguous words are classified.

¹⁰ The appendix of the *Analytical Greek New Testament* explains the tagging philosophy. Additional information is available at <http://www.silverbmt.com/AGNT/index.htm>. The second edition is a more formal classification approach than the first edition, but it still has many functional and unusual classifications. The first edition is Barbara and Timothy Friberg, eds., *Analytical Greek New Testament*, Grand Rapids: Baker, 1981. and the second edition is Barbara and Timothy Friberg, eds. *Analytical Greek New Testament: Greek text analysis*. Vol. 1. Baker’s Greek New Testament Library. Cedar Hill, TX: Silver Mountain Software, 2001.

Morphological Tagging	Characteristics of Tagging System
BibleWorks ¹²	Formal with no functional categories. Adjectives have subtypes, but conjunctions are only coordinate or subordinate.

Here is a simple example of how the morphological tagging system can affect search results. The following chart shows the results of a search for nominative case Greek nouns in the NA27/UBS4 Greek NT with different tagging schemes (all searches were performed in Logos Bible Software):

Greek Morphological Tagging	Nominative Nouns
Logos Greek Morphology ¹³	7305
Gramcord Morphology ¹⁴	7790 ¹⁵
Friberg Morphology ¹⁶	8507
Swanson Greek Morphology ¹⁷	7555

Even though these all use the same Greek NT version,¹⁸ this simple search produces significantly different results due to tagging differences between the texts. The impact of such classification variations would compound in a search with a Boolean (logical) operator that combines one or more nouns as search terms.

¹¹ This is explained in the information window on the Swanson *New Testament Greek Morphology UBS 4th Edition* in Logos Bible Software.

¹² The tagging philosophy of the BibleWorks Greek New Testament is explained in the BibleWorks reference manual.

¹³ The results shown are in the NA27. Logos Greek Morphology is used by numerous Logos Greek New Testament resources with essentially the same text, including NA27, NA28, UBS4, UBS5, Lexham Syntactic Greek New Testament, Lexham Clausal Outlines, and Opentext.org (since Logos 4). Texts that use the Logos Greek Morphology with a substantially different surface text are not considered here, to avoid confusion caused by the different choices of textual variants (e.g. SBL Greek New Testament, Westcott and Hort, Scrivener's Textus Receptus, Stephen's Textus Receptus).

¹⁴ Version 5.5 Alpha on the NA27 text. The NA26 with Gramcord morphology has 7385 nouns, showing that some words were reclassified in the NA27.

¹⁵ The dedicated Gramcord search software finds 3 less instances than Logos does on the same text.

¹⁶ NA27 text.

¹⁷ UBS4 text, which is identical to NA27, except for minor differences in punctuation and paragraphs.

¹⁸ To simplify comparison, only tagged texts based on the UBS4/NA27 are shown. The Byzantine text (Robinson/Pierpont Byzantine Greek New Testament) has 7299 nominative nouns. The difference of 6 nouns compared to the NA27 with Logos morphology is probably due to textual variants in the Byzantine manuscripts, since the same tagging philosophy was probably used.

Several factors account for these differing search results:¹⁹

- **Classifying a word as a noun or an adjective.** For example, in Matt. 4:5, διάβολος (“devil”) is classified as a noun by Gramcord, an adjective by Logos, and a pronominal adjective by Friberg (i.e. as an adjective functioning as a noun). Swanson classifies it both as an adjective and a noun.²⁰
- **Classifying a word as vocative or nominative.** In instances where the vocative and nominative forms are the same, a word could be parsed formally either way.²¹ The nominative might be considered the subject of the verb or a nominative functioning as vocative to address someone. For example, in Matt. 2:6, “land” (γῆ) is classified as vocative by Logos and Swanson, based on the context. Gramcord, however, classifies such words as both nominative and vocative, which allows a search on either case to let the interpreter decide on the meaning.²²
- **Classifying pronouns as a subcategory of nouns.** The Friberg AGNT text classifies “pronoun” as a subcategory of “noun” in 1227 instances. If these pronouns are not included in the search results, this text has 7280 nouns, which is quite close to the Logos count of 7305.²³

Syntactically tagged biblical texts add tags indicating the syntactical function and relationships of words, phrases and clauses. Some syntactic Greek New Testaments in Logos Bible Software include Opentext.org, Lexham Syntactic Greek New Testament and Lexham Clausal Outlines. These syntactically tagged texts enable more complex searches. However, syntactical tagging is more interpretive than morphological tagging and thus search accuracy depends on the syntactical analysis represented by the syntactical tags.

¹⁹ The method of comparison used was: (1) search on both NA27 with Logos Morphology and NA27 with Gramcord morphology; (2) save both searches as verse lists; (3) merge the verse list to show only differences; (4) look at individual highlighted words to see tagging differences. Due to the huge number of examples (more than 7000), only representative verses were examined to find the types of parsing differences that would account for the differing numbers.

²⁰ Although the word can be an adjective or a noun, BDAG only lists 3 instances in the NT where the word is an adjective (meaning “slanderous”). The presence of the Greek article makes the classification as a noun (Gramcord) or pronominal adjective most likely in this verse.

²¹ The nominative and vocative forms are the same for feminine 1st and 3rd declension nouns and all plural nouns.

²² Bethlehem is classified similarly in the verse. Cf. Matt. 3:7: “brood” (γενναμάτα): vocative (Logos, AGNT, Swanson), dual classification: nominative or vocative (Gramcord); Matt. 4:15: γῆ “land” is much debated depending on whether the main influence is the LXX (addressees) or the Hebrew Bible (object of verb): vocative (Logos; Opentext.org, Robinson/Pierpont), nominative (Gramcord, Swanson), dual classification as nominative and vocative (AGNT); John 13:33: “children” (τεκνία) nominative/vocative (Gramcord), vocative (Logos, Swanson).

²³ It is not obvious how to search only for nouns with the AGNT, because the Logos morphology selector pop-up menu does not allow you to exclude pronouns in a search for nouns. However, by manually changing the search from @N?N to @N-N, pronouns are excluded from the search for nouns

3. The Impact of Functional and Unusual Parses

Functional classifications of morphology can produce unexpected search results, particularly if the searcher is unaware of the possible ways a word may be classified. For example, Swanson frequently classifies words as nouns or substantives if they function as a noun (substantively) in context, even when the lemma is technically an adjective in Hellenistic Greek.²⁴ Since it is hard to predict how a program will classify a word in any given passage, the safest approach is to search for all possible classifications of the word and then to manually remove invalid matches.

The BDAG Greek lexicon classifies οὗ as an adverb of place. Gramcord, Logos Morphology and Swanson use this classification of the word. Logos also adds an alternate tag of conjunction 4 times and relative pronoun twice.²⁵ In the majority of occurrences, Swanson also tags the word as a pronoun, which allows searches on either category. By contrast to these largely formal systems, the Friberg AGNT text classifies the word functionally, based on the nuances of the word in the context. It classifies οὗ as an adjective (subcategory adverb, type relative), except for 5 instances in which it classifies it as a conjunction.²⁶

Some of the Friberg functional classifications may be surprising to users familiar with traditional morphological categories. The Friberg AGNT classifies adverbs as “adjectives” with the subtype “adverb”. It introduces a category of “participial imperative” (168 times), although there is no attempt to distinguish other functions of participles. Friberg classifies pronouns as a subcategory of nouns (1227 times). Relative pronouns, however, are generally classified as adjectives with the subcategory “pronominal” (1495), although a few are classified as “adjective – relative” (17) or noun (16).²⁷ In the revised AGNT, Friberg classifies many adjectives that function substantively as “adjectives” with the “pronominal” subcategory. This is not as precise as the “substantive adjective” category used in the original AGNT, since “adjectives – pronominal” also includes many pronouns (particularly relative pronouns). A researcher must be aware of all of these possible classifications to find every occurrences of a particular part of speech.

Functional classifications such as these are more subjective than formal morphological classifications. Their value for searching depends largely on the accuracy of the classifier's interpretation of each passage. While functional classifications appear to offer objective raw data, in fact they represent the prior conclusions of another researcher, which skews the search results to fit the classifier's own viewpoint.

²⁴ Swanson dual classifies 2703 adjectives as a noun (812) or substantive (1891). It is not clear what practical difference is between the 2 categories.

²⁵ Conjunction: Rom. 4:15; 5:20; Cor 16:6; Cor 3:17. Relative pronoun: Matt. 2:9; 18:20.

²⁶ Matt. 18:20; Rom. 4:15; 5:20; 1 Cor. 16:6; 2 Cor. 3:17.

²⁷ Two relative pronouns in Romans 9:21 are classified as “adjective – demonstrative”, but are dual classified as “adjective – pronominal”.

4. Treatment of Classification Ambiguities

Even strict formal tagging schemes sometimes classify words by their function in context, since the parsing of these words is inconclusive based on morphology alone. While in most cases the function of these words is clear in the context, in some instances the classification is subject to scholarly debate. Bible programs are more useful if they mark such words as ambiguous and allowed searching on either classification. Logos, Gramcord and BibleWorks allow searching on multiple morphological classifications where they occur in the text. This allows Bible students to make their own interpretive decisions about verses that are grammatically complex or ambiguous.

For example, the gender of *ποταμοῦ* could be either neuter or masculine. In Matt. 6:13 the meaning is debated: Does the Lord's Prayer ask for deliverance from “evil” (neuter) or “the evil one” (masculine)? Since the Logos NA text classifies the word as masculine, a search for neuter adjectives would not find this verse. The Friberg AGNT and Swanson texts include both parsing alternatives in Logos, so the word would be found with a search under either gender. The Greek NT in the Gramcord program parses this word with both neuter and masculine tags, but for some reason only the neuter tag is visible in the parsing of the Logos version of the Gramcord text.

Gramcord, Swanson, and the Friberg AGNT include multiple morphology tags on some words. Gramcord generally tags more words with multiple classifications than other Greek NT texts and Logos only rarely offers multiple classifications. However, no one text tags all Greek morphology ambiguities.²⁸

Capabilities of Search Software

Although on the surface most Bible programs appear to allow similar searches, there is actually considerable variation in the search capabilities of various programs. In addition, hidden or poorly documented assumptions and limitations of search engines can contribute to surprising search results.

Although all programs include the basic AND, OR and NOT Boolean (logical) operators, many programs lack the sophistication to perform precise searches of complicated grammatical structures. Advanced search features such as word proximity, exclusion of terms, and agreement/disagreement of morphological characteristics enable searches to be fine tuned, which can save the time required to manually remove false matches.

²⁸ For example, a search in Logos for nouns that could be parsed as nominative or vocative finds the following: Gramcord: 340; Friberg AGNT: 10; Swanson: 7; Logos NA27: none. For adjectives that could be masculine or neuter: Gramcord: 15; Friberg AGNT: 51; Swanson: 47; Logos NA27: none. See footnote 25 for an example of Logos alternative tags.

1. Default Boolean Operator

The behavior of Bible programs differs with a search containing multiple words without a Boolean operator. Logos 2 and Bibloai assume an OR Boolean operator, so a search with several words finds verses in which any of the specified words appear. BibleWorks, Gramcord²⁹ and Logos since version 3 assume an AND search, which looks for all search terms anywhere in a verse. To search for a phrase, the words must be enclosed in quotation marks.

2. Lemma or Inflected Form of a Word

The ability to search on the lemma (dictionary form) of a Greek or Hebrew word is essential for finding all occurrences of the word. This is probably the most common search in the original languages of the Bible.

Logos can search on a lemma without specifying the part of speech. This removes the difficulty of finding all occurrences of a word that is classified as different parts of speech depending on its usage in context. By contrast, Gramcord requires entering not only the lemma, but also the part of speech. Thus, to find all occurrences of καὶ with Gramcord, you must search on the word parsed not only as a conjunction (8364), but also as an adverb (832), a particle (3) and even the pronoun καὶ (106), which is a contraction of καὶ and ἐγώ.

Logos is inconsistent about whether a search defaults to a Greek lemma or the inflected form of a word. In Morphological searches, a Greek word is assumed to be a lemma. However, Bible searches look for the specific form of a Greek word, unless the prefix “lemma:” is added to the Greek word.

3. Sensitivity to Diacritical Marks

Some diacritical marks affect the meaning, such as Hebrew vowel points and Greek accents, breathing marks and iota subscripts. It is important to have the option to enter diacritical marks in searches in order to distinguish words that are otherwise spelled the same, such as εἰς (“one”) and εἰς (“into”), which only differ in the Greek breathing mark. However, it is tedious and error prone to always enter accents and other diacritical marks in search expressions. This also can result in missing legitimate matches, since Greek accents can be changed by the context.

In Logos, if Greek is entered without accents or breathing marks, the search ignores diacritical marks. This is convenient for entering searches with a Greek keyboard. However, it is possible to enter accents or breathing marks where they are important to the meaning.

The only time Logos requires correct accents is in searches for a Greek lemma. For example, <Lemma = lbs/el/εἰμι> must be entered (with the accent) rather than <Lemma = lbs/el/ειμι> (without an accent). This is not much of a practical problem, since as you type a search term, a pick list of Greek words appears, so you can select the properly accented word from the list.

²⁹ In this section, “Gramcord” refers to the Gramcord for Windows search software, not the Gramcord tagged Greek New Testament text, which is also available for Logos Bible software.

4. Word Order Sensitivity

For some searches, word order is very important. For example, substantival adjectives require that an article appears one or two words prior to an adjective, not after the adjective or elsewhere in the sentence.³⁰ Similarly, a search for a μέν . . . δέ construction returns false matches if the word order is not strictly controlled, since the reversal of the words changes the meaning.

In other cases, however, it is important to find all permutations of word order. For example, a search for genitive absolutes should allow either the genitive noun or the participle to appear first.

Bible programs differ in the role of word order in search expressions. By default, a Gramcord search specifies the order of the search terms. A search can also allow any word order or define several specific permutations of word order. The Logos Morph Query Builder and Syntax searches also define a specific word order. In BibleWorks and Logos command line searches, by default search terms can appear in any order. Searches can specify the word order with the BEFORE and AFTER operators.

Searches can be further refined by specifying within how many words two search terms must be. Logos, Gramcord, and BibleWorks allow specifying the maximum number of words apart the search terms may be. In Logos, for example, when BEFORE 2 WORDS is inserted between search terms, the first search term must occur 2 or fewer words before the second search term.

5. Duplication of Search Terms

Many grammatical constructions require that the same search term appear more than once (e.g. “δέ . . . δέ or multiple nouns in the same case). Gramcord allows the same term to appear more than once in the search expression. The Logos Morph Query Builder also allows repeating search terms. Logos Syntax searches can repeat a search term, provided that they occur within the same syntactical word grouping.

Logos Bible and Morphology searches, however, cannot repeat a search term with the simple AND Boolean operator. A search for δέ AND δέ finds all verses in which the word δέ occurs at least once. A workaround is to use the BEFORE operator rather than the AND operator: lemma:δέ BEFORE lemma:δέ.³¹

6. Proximity of Search Terms

Many Greek grammatical constructions require that two or more words be in close proximity. A search program should allow specifying the maximum separation of search terms.

³⁰ Of course, the article must agree with the noun in case, gender and number.

³¹ It is essential to search for the lemma, because sometimes δέ is shortened to δ'.

By default, Gramcord assumes search terms are adjacent. It allows specifying the maximum number of words from the beginning to the end of a construction and between pairs of search terms. BibleWorks also allows specifying the maximum number of words between search terms.

The behavior of Logos depends on the search type. For Bible and Morphological searches, multiple search terms must be somewhere in the same *verse*. For Basic searches, multiple words must be somewhere in the same *chapter*. Searches can require that words occur WITHIN, BEFORE or AFTER a certain number of words.³² The Morph Query Builder can specify a span from the first to last search term and the span between pairings of search terms.

One subtle difference in Bible programs is how they count the words in proximity. *Inclusive* proximity includes the first search term in the count, but *exclusive* proximity starts the word count with the word after the first search term. Gramcord uses inclusive proximity, so the smallest possible proximity (*context*) setting is “2,” if there are two search terms. By contrast, Logos Bible and Morphology searches count words exclusively, so adjacent words are “WITHIN 1 WORD.” However, the Logos Morph Query Builder counts words inclusively like Gramcord, so a *span* of “2 words” means the two search terms are adjacent. Thus a Morph Query Builder span of “2 words” is equivalent to “WITHIN 1 WORD” in a Logos command line search.

7. Search Boundaries by Verse, Sentence, Clause

For grammatical searches and many Boolean searches it is important to set the search boundaries by sentence or clause rather than by verse. A program that restricts Boolean searches to a single verse would have difficulty with sentences that span several verses (e.g. Eph. 1:3- 14). For discourse analysis, however, search boundaries would best be set to paragraph or chapter. An ideal program would allow setting search boundaries to a certain number of clauses, sentences, verses, or paragraphs. It would also allow the option of stopping at various types of punctuation marks.

Gramcord and Accordance offer excellent flexibility. The search boundary can be clause, sentence, verse, or paragraph. Accordance also allows boundaries to be by chapter.

Most other programs are more restricted. BibleWorks uses verses as boundaries, but search expressions can cross verse boundaries if the proximity is set to 2 or more verses.³³ Logos shows a different behavior depending on the kind of search. Bible and Morphological searches use the verse as the boundary, but Basic searches use the chapter as the boundary. The Morph Query Builder lets you specify a number of verses as the boundary (“Search Span”), but it does not allow setting a boundary at the end of a clause or sentence. This is a significant limitation for grammatical searches, which focus on the relationship between words within a clause or sentence.

³² Logos 4 and later is a considerable improvement over Logos Libronix 2 and 3. Libronix calculated proximity by the number of characters between words. Proximity by “words” was estimated based on an average of 6 characters per word. Starting in version 4, Logos began measuring proximity by the actual word count.

³³ For BibleWorks, if you search “within 1 verse” this will look at 2 verses (exclusive counting). To only search a single verse set the search to “within 0 verses”.

If there is a conflict between the number of words in proximity and the search boundary, the boundary wins. In Gramcord, if the boundary is the sentence, searches will never cross a full stop (period or question mark), regardless of the number of words in the proximity search setting. If the boundary is a clause, it will never cross a comma, colon or semi-colon (raised dot) to find multiple search terms.

Logos does not allow command line Bible searches that cross verse boundaries (e.g. within a certain number of verses). A workaround is to do a Basic search, which searches for terms within the same chapter. For example, a Basic search for “father WITHIN 30 WORDS son WITHIN 30 WORDS spirit” finds John 3:35-36 and 1 John 4:13-14, which have “father”, “son” and “spirit” within 30 words of one another in different verses. The drawbacks are that the search results are listed by chapter rather than by verse and the search cannot be restricted within a certain number of verses.

Logos Syntax searches can specify boundaries by phrase or clause. This boundary takes precedence over the proximity of a specified number of words between search terms. Logos Morph Query Builder searches will stop searching when the specified span of words or verses is reached, whichever comes first. By default, there is no verse limit so searches will cross verse boundaries to search the required number of words, unless the Span is set to a single verse.

This subtle difference between programs can produce significantly different search results. Gramcord misses 12 examples of sentences with a “μέν . . . δέ” construction because it ends the search at a full stop. In most of these cases, the punctuation mark is a semi-colon, which indicates that the two clauses are closely related. The choice of a semi-colon rather than a comma is often a debatable editorial decision that might vary with different editions of the Greek NT. If the proximity is increased to 2 sentences, more matches will be found, but many false matches will be included.

In Logos, Bible searches miss many examples of “μέν . . . δέ” constructions, since searches are restricted to a single verse. The solution is either to construct a Syntax search or to use the Morph Query Builder with the Span set to 2 verses.

8. Agreement of Morphological Features

Many common grammatical constructions require that certain morphological features agree. For example, a Greek genitive absolute requires a clause with a genitive noun and a genitive participle that agree in gender and number. Conversely, anarthrous nouns require that there be no article preceding the noun in the same case, gender and number as the noun.

In Logos Libronix 3, the Graphical Search allows specifying agreement of morphological features. This allows grammatical searches that are not possible or more difficult to construct in later versions of Logos. Gramcord, Accordance, and BibleWorks also allow great flexibility in agreement and non-agreement of words.³⁴ As a result, these programs can find very complex grammatical constructions with relatively few false matches.

Logos 4 and later allows specifying morphological agreement in Syntax Searches. In Logos 7, the Morph Query Builder was added with the optional Logos Now subscription. Like

³⁴ BibleWorks can even force agreement of morphology in a command line search.

Gramcord, the Morph Query Builder uses a table of search terms which allows easily specifying agreement between any desired search terms. Syntax searches also can find complicated grammatical constructions, but Syntax searches can only find relationships between words within the groupings of words that are specified in the syntactical database, rather than a sequence of words within a sentence. This adds some subjectivity to the searches and requires anticipating all of the ways that words might be grouped by the designers of the syntax database. Since it is difficult to properly design Syntax searches, they tend to be error prone.

9. Exclusion of Search Terms

Many grammatical constructions require that a certain type of word not occur between or before search terms. For example, an anarthrous noun has no article in the same case, gender and number in near proximity before the noun. This search requires the exclusion of an agreeing article before the noun, even if other articles appear that do not agree with the noun. The Granville Sharp rule is another important Greek grammatical construction that requires the absence of an article.³⁵

Many Bible programs are not capable of true exclusion searches. The NOT operator is insufficient, because it omits verses in which the excluded word type appears anywhere in the verse. Exclusion requires a NOTBEFORE operator with the ability to specify a proximity to search terms. Logos command line searches do not have this level of flexibility, although BibleWorks and Accordance have this option in both command line and graphical searches.

For some searches, the lack of a true exclusion operator can be emulated with the NOT operator. For example, to find verses in which the word δέ occurs only once, the search could be: lemma:δέ ANDNOT (lemma:δέ BEFORE lemma:δέ). However, this type of search syntax can yield unintended side effects. For example, the following search will find anarthrous nouns: @N ANDNOT (@N AFTER 2 WORDS @D). This successfully finds verses that have a noun not preceded by an article, but, it does not find *all* instances of anarthrous nouns. Verses such as John 1:1 in are not found (Ev ἀρχῇ ἦν ὁ λόγος), even though ἀρχῇ is an anarthrous noun. Logos searches for verses that meet the criteria within the parentheses (a noun within 2 words after an article). Then it eliminates such verses from consideration, even if there are other word clusters that have a noun without an article in the same verse.

Logos Libronix 3 Graphical searches allow morphological agreement between a noun and the excluded article. It is fairly easy to set up a search for anarthrous nouns. BibleWorks and Accordance both have a similar graphical search design feature.

³⁵ The Granville Sharp construction is article + noun + καὶ + [no article] + noun. The search must exclude an agreeing article before the second noun. The article and both nouns must agree in case, gender and number. The “noun” in the construction can be any substantive, including a participle or adjective. Sharp’s first rule, which is usually simply called the Granville Sharp rule, requires that the nouns be singular personal nouns, but not proper nouns. However, many later scholars have studied other aspects of this syntactical construction, including plural and impersonal substantives. See, for example, Daniel B. Wallace, *Greek Grammar Beyond the Basics: An Exegetical Syntax of the New Testament* (Grand Rapids, MI: Zondervan, 1996), 270-290 and the more in-depth treatment in Daniel B. Wallace, *Granville Sharp’s Canon and Its Kin: Semantics and Significance*. Studies in Biblical Greek 14 (New York: Peter Lang, 2008).

In Logos 4 and later, a Syntax search is the only reliable way to exclude a word before or between search terms. However, such searches can be time consuming and tricky to set up. All possible groupings of search terms must be considered based on the way that words, phrases and clauses are grouped in the syntactical analysis of the particular tagged text. Figure 1 shows the many syntax classifications that must be considered in a simple search for anarthrous nouns.³⁶ Inconsistencies in classifying syntax in biblical texts makes it difficult to be certain that you have actually found all instances of a desired grammatical construction.

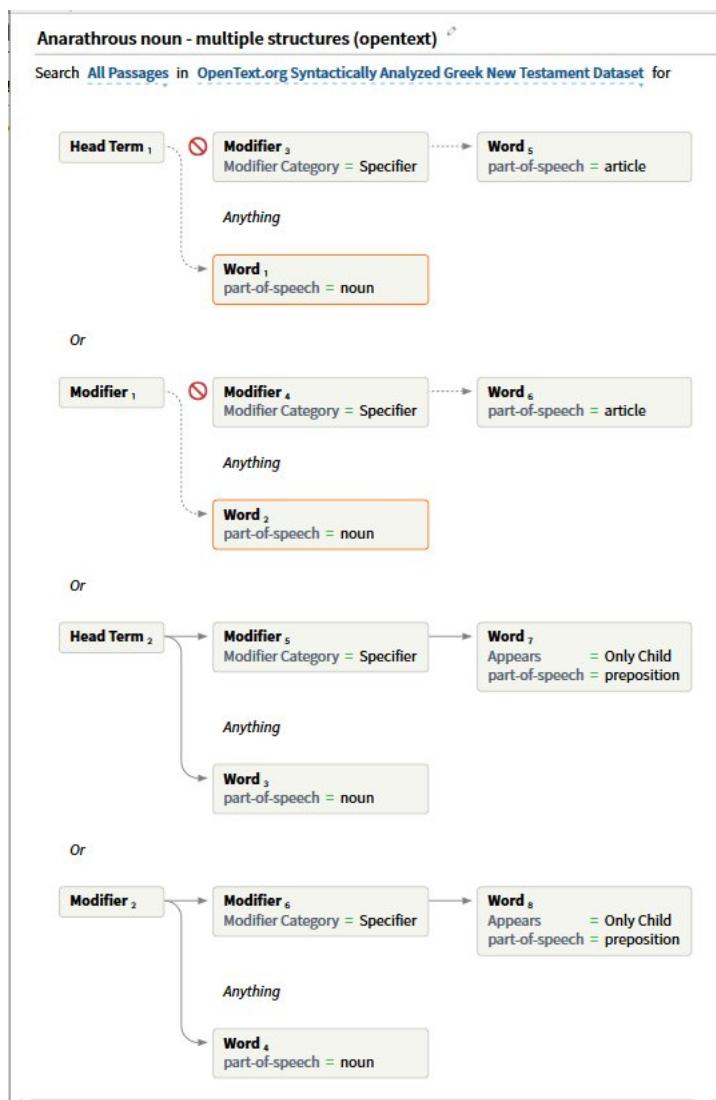


Figure 1: Logos Syntax Search for Anarthrous Nouns

³⁶ See <http://community.logos.com/forums/p/22721/169497.aspx> for a discussion of the various combinations required for a Syntax search for anarthrous nouns.

Gramcord has a much simpler way of setting up such searches through a simple table of search terms. Gramcord allows specifying the required non-existence of a certain part of speech before, after or between search terms. To search for anarthrous nouns, simply specify “article” and the “exclude” option, then enter the number of allowed words between the article and the noun (see Figure 2). Check off the morphological features that must agree between the noun and the excluded article (i.e. case, gender number). Such a search can be set up in less than a minute.

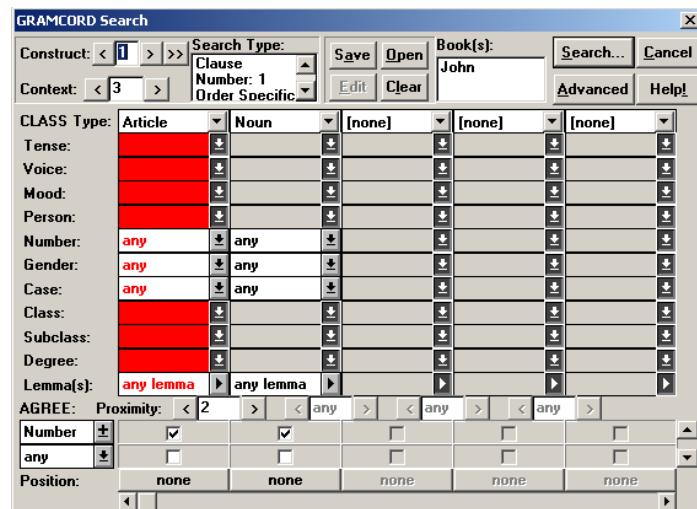


Figure 2: Gramcord Search for Anarthrous Nouns

The Logos Morph Query Builder provides a similar table for designing searches (see Figure 3). Morphological agreement can be specified between any desired search terms, including excluded terms. However, there are a few missing options important for grammatical searches, such as restricting the search boundary to a clause or sentence and specifying the position of a word in a clause or sentence (e.g. first, second or last position).

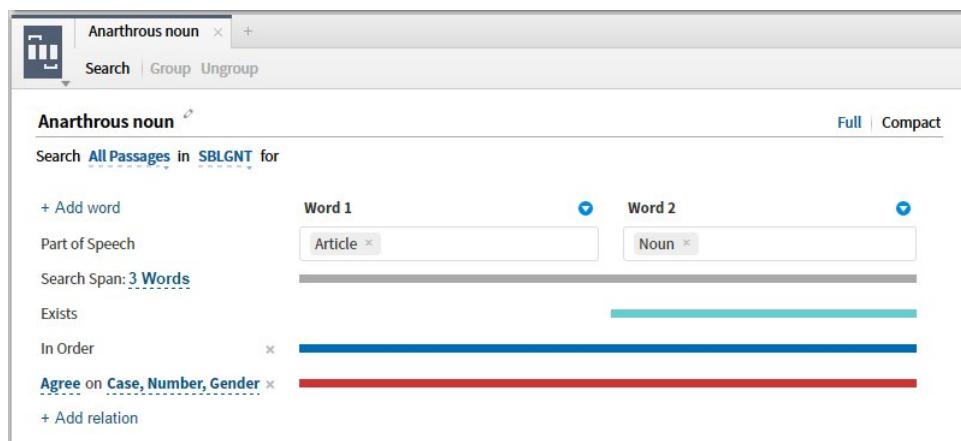


Figure 3: Logos Morph Query Builder Search for Anarthrous Nouns

Although table-based searches are simple to design, the Logos Syntax search has an important advantage. Searches that depend on a sequence of words cannot find words that are grammatically related when they are separated by a great distance in the sentence. Consider a search for substantival participles (a participle governed by an article that agrees with it in case, gender and number). In 1 John 2:4 (ο λέγων ὅτι ἔγνωκα αὐτὸν καὶ τὰς ἐντολὰς αὐτοῦ μὴ τηρῶν, ψεύστης ἐστίν), the article ο governs two substantival participles: ο λέγων . . . καὶ . . . μὴ τηρῶν. The two substantival participles express the compound subject of the main verb: “The one who says . . . and does not keep”. The article governs the adjacent participle λέγων and the participle τηρῶν, which occurs 10 words after the article. Logos and BibleWorks command line searches as well as Gramcord and the Libronix Graphical Search would need a large proximity setting between the article and the noun. This runs the risk of false matches due to the presence of an article that is not grammatically related to a particular participle, even though it morphologically agrees with it.

By contrast, a carefully designed Logos Syntax search should be able to find such an article-participle pair. The designers of a syntactically tagged text such as the Lexham Syntactic Greek New Testament can specify the word(s) that each article governs. Thus a Logos Syntax search for an article-participle pair could find both λέγων and τηρῶν in 1 John 2:4. However, the drawback of a Syntax search is that it depends on the accuracy and consistency of the syntactical tagging on the bible text.³⁷

10. Warning About Incorrect Search Syntax

Logos does not warn if the search syntax is incorrect, such as if there is an unmatched parenthesis or a misspelled operator (e.g. entering NOT rather than ANDNOT or entering NOT EQUALS rather than NOTEQUALS). Similarly if you inadvertently enter “and” rather than “AND” it will search for the word “and” rather than conduct a Boolean search. The search engine takes what you enter literally and returns unexpected results.

An error message or warning would help reduce the number of inadvertent errors due to hasty typing or inadequate knowledge of the search syntax. One suggestion would be to return at the top of the search results “Did you mean...?” followed the search expression with the missing parenthesis or the Boolean operator correctly spelled and capitalized. The user could then click on the link to conduct the intended search.

³⁷ In the Lexham Syntactical Greek New Testament, the participle ποιῶν in 1 John 2:17 is correctly tagged as a substantival participle related to the article ο. But in 1 John 2:4 λέγων is classified as an attributive participle, despite the identical syntax. (The same classification is used for τηρῶν in this verse.) Thus, although the Syntax search allows finding distant agreement of words, the success of such a search depends on the syntactical tagging of specific words, phrases and clauses. This shows that caution must be exercised in the searching on syntactical labels, despite the convenience of using them. The Opentext.org Syntactically Analyzed Greek New Testament only relates λέγων and not τηρῶν to the article in 1 John 2:4, so a search for an article-participle pair would not find τηρῶν in this verse. This illustrates the risk of searches based on syntactical word groupings, which depend on the interpretation of the scholars who created the database.

11. Search Results Statistics

Bible programs vary in the way they report statistics about the number of matches. Logos and BibleWorks report the number of matches in terms of both occurrences and verses. Gramcord only reports the number of matching verses.

With a single search term, the reported numbers are accurate. But if a search involves multiple search terms, the statistics can be misleading in some Bible programs.

Gramcord reports an excessive verse count if the search terms are found in different verses. All verses that are part of the construction appear as separate items in the verse list.

The reporting of the number of occurrences can be even trickier. Ideally, the reported number of occurrences should be based on the number of times the combination of search terms appears. However, Logos command line searches count each occurrence of a search term as a “result”. For example, in a Bible search for μέν BEFORE δέ, Logos reports “3 results” in Matt. 16:14, because μέν occurs once and δέ occurs twice. Although it is true that the search terms occur three times in this verse, there are only two permutations of μέν before δέ, so two is a more meaningful number of occurrences of the construction. With a search that has two search terms and each search term only occurs in one verse, the number of results is twice the number of verses.³⁸ But if any of the search terms occur more than once, it is impossible to determine how many actual combinations of the search terms are found.

When the ANDEQUALS operator is used, the number of results is double what is expected. The ANDEQUALS operator describes two characteristics of a single search term. But Logos treats the two characteristics as if they are separate search terms, rather than two aspects of a single search term.

Logos Syntax searches and Morph Query Builder searches report results in a more meaningful way. The “results” number represents all valid combinations of the search terms. In a search for μέν before δέ, this search reports “2 results” in Matt. 16:14, since there are two combinations of μέν before δέ in the verse.

Incorrectly calculated results can skew the statistics about a construction. The Logos graphical display of Bible or Morphological search results is of less value for comparing relative frequency in various parts of the Bible if the search involves multiple search terms. Logos Syntax and Morph Query Builder searches permit a more meaningful statistical comparison of different parts of the Bible.

12. Search Syntax Quirks

Every computer program has quirks and bugs. A user who is aware of these issues can work around them, but a user that is not aware of them is likely to get incorrect results. Posting a question on a user forum devoted to your software can often help you find ways to live with the quirks.

For example, in Logos it is not obvious how to search for nouns with the Friberg AGNT text. “Pronoun” is a subcategory of “noun” in this tagged text, but the Logos morphology

³⁸ For example, a search for the phrase “καὶ νῦν” reports 26 results in 26 verses. But the functionally equivalent search for καὶ BEFORE 1 WORD νῦν reports 52 results in 26 verses.

selector pop-up menu does not allow you to exclude pronouns in a search for nouns. The work around is to manually change the search syntax from @N?N to @N-N. Replacing the question mark (wildcard) with a dash (exclusion) excludes pronouns from the search.

The Logos ANDEQUALS and NOTEQUALS operators allow you to specify more than one characteristic of the same search term, such as a Greek word that is translated a certain way in a reverse interlinear Bible or a Hebrew word translated by a certain Greek word in the LXX. However, the results with these operators can be unpredictable when searching on morphological tags, unless the morphological search term is put in parentheses. For example, it is helpful to find places where a Greek word is tagged more than one way. But a search in AGNT for the places where a noun is tagged as both nominative and vocative produces no results when the syntax is @N-N ANDEQUALS @N-V. However, if parentheses are placed around each search term the ANDEQUALS operator works properly with morphology. Thus a search for (@N-N) ANDEQUALS (@N-V) finds the desired words that are tagged with these alternative parsings.

Common Errors in Using Bible Software

Even the most powerful search software and carefully tagged biblical text can produce misleading results if the software is not used correctly. There are several common sources of error when using Bible search software.

1. Not Being Aware of the Features and Quirks of the Search Software

If a user does not understand the capabilities and limitations of the search engine, the search may not be formulated correctly. Each program has its own rules of search syntax and search limitations.

Unfortunately, for many bible programs, some important search assumptions are poorly documented, such as some of the quirks noted earlier in Logos searches. For many tagged biblical texts, there is limited documentation of the database tagging guidelines and assumptions. Some subtle search assumptions are documented, yet easily misunderstand. For these reasons, it is wise to experiment with a search that has known results before attempting searches with unknown results.

2. Inadequate Understanding of the Tagged Biblical Text

The better you know the tagging philosophy of the digital biblical text, the better you can search that text. No tagging system is perfect, so it is important to learn as much as possible about the texts available to you.

3. Not Understanding How to Enter the Search Correctly

Many search errors occur when a user does not understand (or forgets) the proper way to enter a particular query. This is a common error with searches using Boolean operators. For example, Bible programs vary in the syntax for Boolean NOT searches: NOT, ANDNOT, AND NOT, or a dash are used in various programs. In Logos, if a user enters NOT rather than

ANDNOT, the search will look for the presence of the word “not”. The search appears to succeed, but has invalid results.³⁹

Many beginning users of Bible software assume that they can simply enter a phrase to search for that phrase. But in Logos, this finds verses in which all of the words appear in any order, regardless of how far apart they are. Each program has a particular syntax to search for a phrase. In Logos 4 and later, the phrase must be entered in quotation marks.

Since Greek words change their spelling (morphology) depending on their usage in the sentence, one must search on the lemma (dictionary form) of the word in order to find all occurrences of the word. The most common error of beginning users of Bible software is to search for the form of the Greek word found in a given sentence in the NT, which will miss many occurrences of the word. Since some results will be returned, it is not always obvious that the search was incorrectly formulated.

4. Inadequate Understanding of the Grammatical Construction

A user may not understand the grammatical construction well enough to formulate a search accurately. It is wise to read about a construction in Greek grammar books before attempting a computerized search, so that the search may be formulated more precisely. This will also provide a sense of the reasonableness of the search results.

For example, a Greek future perfect periphrastic construction (e.g. Matt. 16:19 “will have been loosed”) consists of a future form of εἰμί (“will be”) and a perfect participle in the same clause. When searching for Greek future perfect periphrastics, it is not enough to find future forms of εἰμί within a certain number of words of a perfect participle. The two words must be in the same clause and they must function together as a single grammatical unit in the sentence. Restricting the search to a single clause and excluding intervening finite verbs will reduce the number of false matches. Such a search can be facilitated by a Syntax search in Logos or by setting the search boundary to a single clause in Gramcord.

5. Failure to Search for all Permutations of the Construction

Some searches are word order specific, such as an article preceding a noun. Other searches allow multiple arrangements of the search terms, such as Greek attributive adjectives, which permit 3 specific arrangements of words.

It is easy to assume erroneously that a single search has found all examples of a construction. A thorough search must consider all valid orders of the search terms. For example, in a Greek genitive absolute construction, the noun can precede or follow the participle. Logos Syntax searches and Gramcord allow a single search to include several different search permutations. With Logos command line morphological searches, multiple searches are required if the BEFORE or AFTER operators are used.

³⁹ Entering “NOT” in a Logos search will find verses where Greek words such as οὐ, μή and δε are used, because they can be translated “not”.

A thorough search must also find constructions with functionally equivalent parts of speech. For example, the Granville Sharp construction can use substantival participles or adjectives instead of nouns.

6. Not Manually Eliminating False Matches

Complicated searches will sometimes produce false matches which must be manually eliminated. Each match must be examined in context to verify that it is a legitimate example of the desired construction.

In Logos, the search results must be saved to a passage list before false matches can be deleted. Unfortunately, this loses the highlighting of search terms, which makes it harder to detect the false matches. It would be more helpful if false matches could be manually deleted directly from the search results list, as Gramcord allows.

Tips for Accurate Bible Searches

If your goal is to find all instances of a particular construction, several strategies can improve your results:

1. **Start by searching a portion of the Bible that includes known examples of the desired construction.** Verify that your search finds legitimate examples of what you are looking for. When you are confident that your search works properly, extend your results to the rest of the Bible.
2. **Verify that your results include known examples of your construction.** For example, when I was searching for anarthrous nouns, the absence of John 1:1 showed me that there was something wrong with my search, even though other legitimate examples were found.
3. **Verify that your results are reasonable.** If there are not too many results, you can double check each match. At the least spot check to see if the results are really what you were looking for.
4. **Construct a complex search one element at a time.** Test the search with one search term and then gradually add additional elements. This let you verify that you are getting reasonable results and lets you catch errors and assumptions.
5. **Save your search criteria periodically.** This saves time when reusing or editing a complex search criterion and lets you revert to a working version of the search if a change breaks your search. With command line searches in Logos, you can reuse previous searches from the search history. Logos Syntax and Morph Query Builder search documents can be named, so they can be edited or reused later.
6. **Be sure you have selected the correct Bible text.** If your software includes several versions of these Greek texts, pick one that has the tagging suitable for your task. For example, BibleWorks includes both tagged and untagged Greek NT texts, but lemma searches will only work properly with a tagged text.

7. **Choose one Bible text to work with regularly.** This lets you learn the features and quirks of the text, so you will be able to do better searches on it.
8. **Learn about your digital text.** Learn about its tagging philosophy and assumptions. How does it handle ambiguous parsings and functional classifications? Are there any known tagging errors or unusual classifications?
9. **Understand the limits and capabilities of your search engine.** Is it sensitive to word order or letter case? Will it search past periods or verse boundaries? Test your program on known searches to understand it better. Many subtleties are not documented and can only be learned through experience.
10. **Consult the user forums for your Bible software to learn about surprising search results.** There is a good chance someone can help you find the error in your strategy or a work around for the limitations of your search software.
11. **Study the grammatical construction before searching with a Bible program.** Read about it in a Greek grammar book or Bible commentary to understand what you are looking for. This will help you formulate your search to find all possible permutations of the construction and will help you verify whether your results are reasonable.
12. **Search for all possible permutations of the construction.** A thorough search must consider all valid orders of the search terms. For example, Greek genitive absolutes could have the genitive noun preceding or following the corresponding genitive participle. A thorough search must also find instances with functionally equivalent parts of speech. For example, if you are looking for nouns, would a Greek adjective or participle that is functioning as a noun also be acceptable?
13. **Manually eliminate false matches.** Even the best software will sometimes produce false matches which must be manually eliminated. Some false matches can only be determined by understanding the meaning of the construction in context.⁴⁰
14. **Practice on searches with known results to learn how your program works.** Searches for grammatical features and other complicated structures involve the interaction of many subtle parameters, many of which are learned from experience.

Conclusion

“Can I trust my Bible software?” The answer largely depends on how well you know the software. Any tool is only as good as the craftsman who is using it. The more you use the tool, the more confidence you will develop in its reliability. Bible programs are powerful tools, but they contain many pitfalls awaiting the unwary researcher.

⁴⁰ For example, in a search for future perfect periphrastics in Gramcord or with the Logos Morph Query Builder, Luke 1:45 and 6:40 can only be eliminated by hand, since the fact that the participle functions as a substantive is only indicated by the meaning in context. Gramcord finds fewer false matches due to the ability to set the search boundary to a single clause.

In part the answer to this question depends on the purpose of your search: Would you be content simply to find a few valid examples of a construction or do you want to find every occurrence in the Bible? All Bible students need to be able to trust that their software produces legitimate examples of what they are looking for. Scholars need comprehensive search results, so accurately tagged biblical texts and precision search tools are essential for their work.

Bible software can save considerable research time compared to using printed concordances and other reference books. However, to get the most reliable results, researchers should know the capabilities, design assumptions and limitations of their search software and digital biblical texts. When used wisely and carefully, these modern research tools can enable Bible students to dig more deeply into the Scriptures.