CHAPTER 6

Using Natural Language Processing to Search for Textual References

Brett Graham

1 Introduction

In natural languages, as opposed to computer languages like C or Pascal, the words and syntax are not artificially defined; instead, they develop naturally. Typical examples of natural languages are those that are spoken in human communication, such as the English, French, and Japanese languages. However, the term natural language can also refer to written text, such as Facebook postings, emails or even text messages. As well as changing over time, natural languages also vary among different cultures and people groups. So, for example, the words and syntax that a teenager might use to write a text message on their phone are likely to be different to the words and syntax that Shakespeare used to write Othello.

Within computer science, the term Natural Language Processing (NLP) refers to way computers are programmed to understand natural languages. At a basic level, NLP involves three steps – lexical analysis, syntax analysis, and semantic analysis. The complexity of each of these steps is perhaps best illustrated through looking at how three well-known programs incorporate NLP; namely, Microsoft Word, the Google search engine, and Apple's Siri.

If you were to type (or copy and paste) the following string – "Can I be worn jeens to church?" – into Microsoft Word then it will perform simple lexical analysis by grouping the characters into tokens (i.e. words) using the whitespace and punctuation as separators. Having done this, the program will then consult its dictionary and recognize that "jeens" is not a valid entry. As a result, it will place this word in red, somewhat like this:

Can I be worn jeens to church?

If Microsoft Word's NLP was very intelligent, it would be able to detect that "jeens" is a misspelling of "jeans" and then automatically change the spelling for you.[1] Having made the correction, Microsoft Word will then recognize that

1 Word allows you to train it to do this correction via the Tools->Autocorrect menu option.

all the tokens in the string are valid, allowing it to move on to syntax analysis. It performs this analysis based on its understanding of English grammar, which is typically represented as a set of syntax rules. The sections of the string that violate the syntax rules are then placed in green and so the string will now look like this:

> Can I be worn jeans to church?

If Microsoft Word's NLP was even more intelligent then it would attempt to autocorrect the grammar of the sentence for you by scanning through all its syntax rules to find the closest match. In this case, it is likely to be a rule that looks like this:

> <question> = <adverb> <subject> <verb> <object> <prep phrase> <question mark>

It might then change the sentence for you in order to conform it to this closest rule, as follows:

> Can I wear jeans to church?

As complicated as this might sound, the next step, semantic analysis, is even harder. Such analysis would involve the program trying to understand the meaning of the sentence. Subsequently, the program might be intelligent enough in order to suggest a more formal alternative, such as:

> What type of clothing is appropriate for church?

While Microsoft Word is not yet able to perform such complicated analysis, the Google search engine attempts to do this when it responds to a user's request. For example, if you were to copy and paste the string "Can I wear jeans to church?" into this search engine, it will attempt to find the answer to this question by searching for web pages that might be relevant. Typically, the set of matching pages will be large, so the search engine will rank the results by its own criteria, such as, "most matching keywords", or "most recently uploaded", or "most visited page" etc. At the time of writing, the highest-ranking answer (i.e. one at the top of the displayed list) is an entry from <http://www.wikihow.com/Dress-For-Church-Services>, which reads as follows:

> Black dress pants are the best option for a person attending a church service. If you don't have a pair, you can wear clean and wrinkle free casual slacks or khakis as an alternative. Avoid shorts.... And [if] you do wear jeans, do not wear ones with patches or holes.[2]

Although the response does appear to answer this particular question, it only works because the same question has been asked before and the answer has been posted on the Internet. As such, the search engine is not really working out the answer; rather, it is relying on existing research. This means that the search engine's semantic analysis will only ever work for old questions, not new ones. This is demonstrated by the fact that if you ask it a question that has not been asked before, such as, "Can I wash jeans at church?" then the search engine is unable to find an appropriate answer. Instead, it simply returns the set of Internet postings that it thinks are the best matches for the words in the question, which in this case turns out to be a similar set of pages to the first question.[3] Thus, the search engine is not doing true semantic analysis (because this task is beyond the limits of current computing); it is just approximating semantic analysis to the best of its ability.

While Microsoft Word and the Google search engine perform NLP on written (or typed) text, the latest generation natural language processors are able operate on spoken text.[4] These include programs like Apple's Siri,[5] as well as Amazon Alexa[6] Google Assistant,[7] and Microsoft Cortana.[8] The reason why such technology is so popular is that it has the potential to answer almost every question and perform almost every request. That is, not just existing questions, but new ones as well! Not only can you ask, "How far is it to the moon?" but you also say, "Please order me a pizza." This is because Siri is not simply looking at web postings on the Internet (though it can do this) but it is asking other computer programs to perform actions on its behalf. These programs can either be running on the same device/computer or they can be running on other platforms across the Internet. In computer science, this second type of program is known as a "web service". In order to answer new questions (i.e. questions that

2 This is the response of the Google search engine on June 3, 2017.

3 This is the response of the Google search engine on June 3, 2017.

4 The application Google Assistant allows the Google search engine to have a voice activated interface. See, "Google Assistant", *Wikipedia*, May 17, 2017 <https://en.wikipedia.org/wiki/Google_Assistant>, accessed on 10.04.19.

5 "Siri", *Wikipedia*, May 18, 2017, <https://en.wikipedia.org/wiki/Siri>, accessed on 10.04.19.

6 "Amazon Alexa", *Wikipedia*, May 18, 2017, <https://en.wikipedia.org/wiki/Amazon_Alexa>, accessed on 10.04.19.

7 "Google Assistant", *Wikipedia,* May 17, 2017, <https://en.wikipedia.org/wiki/Google_Assis tant>, accessed on 10.04.19.

8 "Cortana (Software)", *Wikipedia*, May 31, 2017, <https://en.wikipedia.org/wiki/Cortana>, accessed on 10.04.19.

do not have their answer posted on the Internet), such as, "When is Peter's birthday?" programs like Siri might consult another application, like Apple Contacts, in order to find the birthday field of Peter. For more complicated requests, such as "How many recent Facebook postings are influenced by Shakespeare?" where there is no relevant Internet posting, Siri would need to ask a web service to do the research for it.

This paper explains how these recent advances in NLP technology can be harnessed to search for allusions and influences to ancient texts. We will begin by investigating the variety of reference forms that were used in ancient literature. Subsequently, we will analyze the recent projects in the Digital Humanities with the aim of determining how effective these projects are in detecting the different reference forms. After recognizing the similarities of these projects, the paper proposes a generic NLP algorithm for detecting textual references. The algorithm is designed to be generic so that can be used to detect any type of textual reference in any type of text (or even a oral allusion to an oral speech). It is suggested that the best way to implement this algorithm is as a web service so that it can be invoked by any Internet search engine, like Google's, or by any virtual personal assistant, like Siri or Cortana etc. This implementation would suit the intention of the algorithm, which is to answer new questions that have not being asked before.

After briefly explaining how this new algorithm works, the final section of this paper will describe how it can benefit biblical (and other textual) studies. The most significant of these benefits is the ability to highlight potential references that are not found by other models, whether automated or manual. In this regard, several examples will be given from applying the algorithm to the Pastoral Epistles. Furthermore, the new algorithm also shares several of the benefits that have been highlighted in recent attempts to automate the detection of textual references. In particular, computers can not only broaden the scope of which documents are searched, but they can also gather metadata from these searches, such as which source texts that a particular author was more inclined to reference, or when and where a particular source text has been most influential in history.

2 The Variety of Reference Forms

The onset of the digital age has brought with it the potential to automate the search for textual references, thereby allowing large databases of source texts to be quickly scanned. However, in order to find as many references as possible, it is important to know exactly what to look for. Therefore, this present section catalogues some of the different reference forms (i.e. the various ways that words are borrowed) in ancient literature.

It has been said that one of the characteristic features of the work of Clement of Alexandria is "the presence of borrowed material [...] taken more or less accurately from other authors' and "culled from every nook and cranny of the nearly thousand-year span of Greek literature."[9] The difficulty of identifying this "borrowed material" is compounded by the fact that he rarely acknowledges his sources; instead "most of the time Clement connects a thought from outside by no more than a single word, a brief formula, a hidden allusion or a mere hint."[10]

While this Early Church Father may be an extreme case, Clement's habit of borrowing from previous literature was certainly not unusual. The Jewish and Christian Scriptures, for example, contain numerous links to earlier texts, both in the form of marked citations as well as un-marked parallels and echoes of their predecessors.[11] Meanwhile, students of the ancient rhetorical schools were explicitly encouraged to embellish their writings with quotations from and allusions to famous authors. According to Quintilian, this involved imitating "the practice of the greatest orators, who [appealed] to the poems of the ancients [...] for the support of their arguments" (Inst. 1.8.10).[12] Likewise, Philo is said to have "borrowed" from a large number of other authors including the Greek philosopher Plato;[13] and the works of Eusebius of Caesarea have been described as "a rich mine of fragments of Greek literature."[14]

Not only was borrowing from earlier works commonplace, there was also a variety of ways in which this was done. Apart from the authoritative quotations that were encouraged by the rhetorical schools, a number of other methods were also employed. Many of Philo's references, for example, are paraphrases rather than quotations, based apparently from memory rather than from a physical text.[15] Similarly, the hymns of Qumran (known as the

9 Van den Hoek, Annewies, *Clement of Alexandria and His Use of Philo in the Stromateis: An Early Christian Reshaping of a Jewish Model*, Leiden: E.J. Brill, 1988, 1.

10 Van den Hoek, *Clement of Alexandria*, 1.

11 Hays, Richard B., *Echoes of Scripture in the Letters of Paul*, New Haven: Yale University Press, 1989, 14. For support of this claim, see "Appendix IV – Loci Citati Vel Allegati" in: *Nestle-Aland Novum Testamentum Graece* (28th edition), Aland, Barbara, et al., eds., Germany: Deutsche Bibelgesellschaft, 2012. See also, Louden, Bruce, *Homer's Odyssey and the Near East*, Cambridge: Cambridge University Press, 2011. Louden finds evidence to suggest that the Jewish and Christian Scriptures also contain 'allusions' to Homer's Odyssey.

12 Quintilian, *Institutio Oratoria Books I-III with an English Translation by H.E. Butler*, Cambridge, Massachusetts: Harvard University Press, 1920), 151.

13 Runia goes as far as saying that 'one can read the whole of Philo's works without coming across a single original thought', Runia, David T., *Philo of Alexandria and the Timaeus of Plato*, Leiden: E.J. Brill, 1986, 9.

14 Van den Hoek, Annewies, *Clement of Alexandria*, 1.

15 Runia, David T., *Philo of Alexandria*, 369. Likewise, Hartog says of Polycarp, "[his] habit of loose quotation demonstrates that he usually quoted from memory and that he felt free

Hodayot) refer to other texts either by summarizing their ideas and themes or by drawing structural parallels.[16] Likewise, the so-called "Testamentary Literature" seeks to gain acceptance by imitating the structure of Jacob's last word (or testament) to his sons.[17] Different again is the book of Jubilees, which interweaves short phrases and groups of verses from the narrative of Genesis and Exodus with "extensive material from other books, in the form of quotation, but also, and more frequently, allusion."[18] A similar approach is adopted in the Prayer of Manasseh, which alludes to the events of 2 Chronicles 33, as the following comparison illustrates:[19]

2 Chronicles 33	Prayer of Manasseh
[Manasseh] … provoking his [Yahweh's] anger	I provoked your fury (or anger)
	I set up idols
… placed … the idol … in the Temple.	I am ensnared,
… Manasseh with hooks	I am bent by a multitude of iron chains
… in chains …	
humbling himself deeply	I am bending the knees of my heart
before the God of his ancestors.	before you, God of our fathers.

These few examples highlight the variety of reference forms that were used in ancient literature. Not only were citations and quotations common, but more subtle references such as paraphrases, keywords and structural parallels were also used. The following table summarizes these reference forms.

to creatively edit his sources". Hartog, Paul, *Polycarp and the New Testament the Occasion, Rhetoric, Theme, and Unity of the Epistle to the Philippians and Its Allusions to New Testament Literature*, Tübingen: Mohr Siebeck, 2002, 172.

16 Hughes, Julie A., *Scriptural Allusions and Exegesis in the Hodayot*, Leiden: Brill, 2006, 51. Hughes also makes a helpful distinction between allusions and mere "coincidences in vocabulary": the former being *intentional* references to *specific* texts, whereas the latter are unconscious repetitions of Scriptural language that were also adopted by the wider Qumran community.

17 Charlesworth, James H., "The Pseudepigrapha as Biblical Exegesis", in: *Early Jewish and Christian Exegesis: Studies in Memory of William Hugh Brownlee*, Evans, Craig A., Stinespring, William F., eds., Atlanta: Scholars Press, 1987, 139-152, 145.

18 Crawford, Sidnie White, *Rewriting Scripture in Second Temple Times*, Grand Rapids: William. B. Eerdmans Publishing Company, 2008, 64. Jubilees has the rare distinction of being found at Qumran as well as being preserved through Christian scribes, especially via the Abyssinian (Ethiopian) Orthodox Church which granted the book canonical status.

19 This example is noted in Charlesworth, James H., "The Pseudepigrapha as Biblical Exegesis", 144.

TABLE 6.1 The different reference forms; ©BRETTGRAHAM

Reference Form	The Way that the Text is Borrowed
Quotation	Verbatim
Paraphrase	Re-wording of a single clause
Single Keyword	One word
Multiple Keywords	Words from multiple clauses that are copied to a single clause
Structural Parallel	Words from multiple clauses that are copied to multiple clauses

This above table serves to highlight that the references come in a variety of forms. This list of reference forms is unlikely to be exhaustive; it is, however, illustrative in that it reveals the complexity involved in trying to detect every reference.

3 Combining Modern Computers and Ancient Texts

In the past, the task of identifying textual references was the domain of individual scholars who each manually searched the set of source texts that he/she was familiar with. However, the onset of the digital age has meant that searches can now be performed on any source text, whether familiar or not, at the click of a button. Event-driven software programs, like Accordance and Logos, and Internet search tools, like Bible Gateway and the *Thesaurus Linguae Graecae* (TLG), enable biblical scholars to perform word searches at a much faster rate than the traditional paper-based approach. For a complex query with multiple search words, it usually takes more time to type in the search string than for the computer to return the result. This difference becomes even more pronounced when large numbers of searches are involved, such as when searching for allusions and influences across an entire document. Thus, over the last decade several projects in the Digital Humanities have attempted to overcome this problem by introducing a level of automation to the generation (and running) of searches. These projects usually involve the adaptation of algorithms that are commonly used in NLP, including the Bag-of-Words, Greedy String-Tiling, and Sequence Alignment algorithms.

The Bag-of-Words algorithm uses a "hashing function"[20] to convert a line of text into a vector (represented as a set of numbers), where the entries of the vector contain the number of occurrences of each different word in that line.[21] Two lines of text can then be compared based on the similarity of their vectors,[22] with those above a specified threshold being marked as related in some manner, such as one being a paraphrase of the other. The vector entries (i.e. the word counts) are indexed based on the value returned by the hashing function rather than their original order in the text, making this algorithm particularly useful for languages such as Greek where the word order can vary, or when the borrowed text has been paraphrased or modified through word insertions/deletions. The Bag-of-Words algorithm has already been used effectively to study the possible use of Mark in Luke's Gospel.[23]

The Greedy String-Tiling algorithm divides a source string (such as a line of text) into "tiles"(i.e. sequences of words) and then places them on top of a target string in places where the words match.[24] Then, if a specified percentage of the target string is covered with tiles, one string might be dependent on the other. This algorithm is described as "greedy" in that it tries to make each tile as big as it can be, even though two smaller tiles placed in the same location might cover more words in total. Hence the algorithm may not always achieve the best solution (since not all permutations are considered), but it finds a good result in a shorter timeframe. The Greedy String-Tiling algorithm is used in the METER project at the University of Sheffield[25] and is most effective when the words are borrowed in sequences (such as in quotations).

The third type of algorithm, Sequence Alignment, divides the source and target strings into overlapping sets of consecutive words. These sets are called "shingles"or "n-grams", where n is the specified number of words in each set

20 The hashing function converts each word into a number, or index, which is then used to order the words. An example of a simple hashing function is to sum the unicode of each letter in the word.

21 The algorithm can also be applied to a sentence or a verse, or even to a whole document. It is commonly used to filter messages in email accounts by defining a vector for a typical spam message (such as having one or more occurrences of the word "Viagra") and comparing this with the vector of each new message.

22 This comparison is made using a "cosine measure". See Dale, Robert, Moisl, Hermann, Somers, Harold, eds., *Handbook of Natural Language Processing* (1st edition), New York: CRC Press, 2000, 471; Lee, John, "A Computational Model of Text Reuse in Ancient Literary Texts," *Proceedings of the 45th Annual Meeting on Association for Computational Linguistics*, Prague, Czech, June 23-30, 2007, 472-479, 475.

23 Lee, John, "A Computational Model of Text Reuse in Ancient Literary Texts".

24 Tiles, or words, that do not match are discounted.

25 Clough, Paul, et al., "METER: MEasuring TExt Reuse," *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL'02, Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, <http://eprints.whiterose.ac.uk/78530/>, accessed on 10.04.19..

and is usually two or three words ("bigrams" or "trigrams").[26] The comparison of the two strings begins by searching for one common n-gram between the source and target, after which the surrounding context is searched for other matches. Then, if a minimum number of n-grams are found together within a maximum distance of each other (or context), the algorithm signals a potential dependency.[27] Like Greedy String-Tiling, the Sequence Alignment algorithm works best when the words are borrowed in sequence, such as quotations, but it can also be used to detect paraphrases if n is small (e.g. one or two). This flexibility has contributed to its use in several recent studies, including being used to search for text re-use across the TLG database.[28]

In summary, there have been a number of recent projects in the Digital Humanities that have sought to detect textual references across a particular set of source texts. In essence, each of these projects is a natural language processor that automates the first two steps of NLP, lexical and syntax analysis. Due to the complexity of NLP, these projects involve written (rather than spoken) texts where the lexical analysis can be largely pre-determined. Each of these projects has a slightly different form of syntax analysis (i.e. determining which combination of words could be valid reference) that is an adaptation of three types of algorithms (as mentioned above), all of which are aimed at detecting verbal similarity. Only potential references with a minimum level of verbal similarity are passed on for semantic analysis (which is done manually on account of the complexity of this task). This approach is inclined to prefer quotations and paraphrases as opposed to single keywords. The syntax analysis can be configured to detect single word matches but that would mean all such matches would need to be passed to the manual semantic analysis phase, which is not practical for large databases of texts.

Furthermore, these programs assume that the syntax of references (i.e. the arrangement of the borrowed words) is static. However, the form of textual references in poetry, for example, need not be the same as those in prose. Likewise, the way that the Early Church Fathers make reference to the New

26 So for example, the string "Shakespeare wrote many plays about love" would be divided into four trigrams: "Shakespeare-wrote-many", "wrote-many-plays", "many-plays-about" and "plays-about-love". Some adaptations of this algorithm omit certain classes of words, such as articles and prepositions. See, for example, Olsen, Mark, Horton, Russell, Roe, Glenn, "Something Borrowed: Sequence Alignment and the Identification of Similar Passages in Large Text Collections", *Digit. Stud. Champ Numér. 2/1*, May 17, 2011 <https://www.digitalstudies.org/articles/10.16995/dscn.258/>, accessed on 10.04.19.

27 Both the minimum number of n-grams required, or "span", and the maximum separation between them, or "gap", are configured as parameters of the algorithm.

28 Büchler, Marco, et al., "Unsupervised Detection and Visualisation of Textual Reuse on Ancient Greek Texts", *J. Chic. Colloq. Digit. Humanit. Comput. Sci. 1/2*, June 16, 2010, <https://knowledge.uchicago.edu/record/133>.

Testament is likely to be different to the way that way that modern pop music does. As such, reference forms are not fixed but context dependent. Therefore, the following section outlines a generic NLP algorithm that is designed to detect references in a variety of contexts.

4 A Generic NLP Algorithm

A generic algorithm for detecting text references using NLP is simple; it is just three basic steps:

Lexical analysis,
Syntax analysis,
Semantic analysis.

The goal of the first step, lexical analysis, is to parse a target text in order to produce subsets of words that are then passed on to the syntax analysis phase. For a traditional natural language processor, each subset is a sentence as delimited by punctuation. However, the "maximum subset size" is treated as parameter of the algorithm because the optimum value of the parameter is dependent on the target text. For example, the optimum value for detecting how Ancient Greek literature makes references to other Ancient Greek literature will be different to the optimum value for detecting how modern English novels refer to the same literature. Thus, the optimum value for the parameter for a particular type of target text is learnt rather than being fixed. This process of learning the optimum value is achieved by training the algorithm to detect known textual references that occur in target texts of the same type.[29]

The lexical analysis phase also involves parsing each word of the target text[30] and then determining the lexical alternatives of the words.[31] These alternatives are a set of one or more words in the language of the source text that can match the word in the target text during a search. They are used in the syntax analysis phase to find potential references between the texts. The target and

29 This training of the algorithm is explained further below.

30 If the source texts were not previously parsed, these would need to be processed as well. The parsing involves finding the root word and determining the lexical form of the word. The recently released natural language processors, like Siri or Cortana, are able to perform this type of lexical analysis on a spoken target text.

31 While the parsing of the target text might need to be performed at run-time (unless it is an existing written text), the lexical alternatives for each word in a language can be preprocessed and then retrieved at run-time.

source texts can be in different languages (such as the target texts being in Ancient Greek and the source texts being in Hebrew) and the lexical alternatives make the translation between the languages possible.

The algorithm has a parameter called "level of alternatives" that is used to fine-tune the search process for each different reference form. This parameter has three different values: "one word,"[32] "same root,"[33] and "synonyms." The level that is selected will influence the amount of matching source texts for each search. When the source texts are large, setting this value to "synonyms" will make it difficult to find a combination that is unique to one passage (or even potentially unique). Therefore, in testing the algorithm (see Section 4 below), the value of "level of alternatives" for the keyword reference form was set to "one word' and value for the other reference forms was set to "same root."

Syntax analysis on each subset of words involves scanning through all the syntax rules to find the rule (or set of rules) that matches the words. If the algorithm were implemented as a web service, these syntax rules could be passed to the algorithm through a file (or data stream) in a standard format called XML. Like the syntax rules of a traditional language like English, the syntax rules for references are relatively simple, but the number of rules is quite large. Every form of reference (like the ones listed in Table 6.1 above) needs several rules for every different length subset. Like the "maximum subset size" parameter, the "syntax rules" are a parameter of the algorithm in order to make them context dependent. This feature allows the algorithm to be used to learn the most effective set of rules for each particular context.

When a matching syntax rule is found, the potential reference is passed on to the semantic analysis phase. While a syntax rule defines a legitimate way that the words can be arranged in order to form a reference, not all of these arrangements will constitute an actual reference (i.e. the semantic analysis will reject potential references that are not meaningful). Due to the current limitations of semantic analysis (i.e. it needs to be done manually to be truly accurate) the algorithm only investigates syntax rules where the matching words are rare.[34] This is based on the assumption that intentional references

32 When the target and source texts are in the same language, the 'one word' is the same word. For differentlanguages, the "one word" is a word from the source language that is commonly translated as that word in the target language.

33 For example, words on the same root of σῴζω ("I save") are σωτηρία ("salvation"), σωτήριος ("saving"), and σωτήρ ("savior"). Where appropriate, some very common synonyms might also included amongst a word's 'same root' lexical alternatives, such as κύριος ("lord") as an alternative for θεός ("God").

34 This is analogous to the logic of Inverse Document Frequency (IDF).

usually appeal to particular passages.[35] To achieve this goal perfectly, the borrowed words in the target text would need to be "perfectly singular". That is, the words would only be found together in just one source text – the text being referred to. However, not all references achieve this perfect singularity.[36] For example, calling someone a "good Samaritan" was originally meant to be a reference to the kind person in Luke 10:25-37, but the word "Samaritan" (Gk. Σαμαρίτης)[37] is found in seven passages of the New Testament, two of which contain someone who could be considered "good".[38] So, while a reference might intend to refer to single passage, in some cases this goal will not be achieved.

Therefore, the algorithm uses a parameter called "potential singularity" as a way of identifying which of the syntax rules to perform semantic analysis on. Like the other parameters, the optimum value for potential singularity is context dependent and needs to be learnt for each type of target text.

```
The algorithm is summarized below:
Parse the target text into subsets of words
For each subset
    Scan the syntax rules
    For each matching rule
        If the words are potentially singular
            Perform semantic analysis
```

The following section now describes how this algorithm was tested in the field of biblical studies.

35 Perri, Carmela, "On Alluding", *Poetics 7*, 1978, 289-307.This paper refers to "intentional" references since the task of NLP is essentially to determine the speaker's/author's intended meaning. The algorithm that is presented in this paper will actually detect both "intentional" and "unintentional" potential references, but only those where the matching words are rare, or "potential singular". This limitation is based on the theory of allusions as described by Perri and is used to reduce the amount of semantic analysis that is required by the algorithm.

36 Contra Hartog and Kittel who require perfect/absolute singularity. See Hartog, Paul, *Polycarp*, 174; Kittel, Bonnie Pedrotti, *The Hymns of Qumran* Ann Arbor, Michigan: Scholars Press, 1981, 51.

37 The word "good" is not found in Luke 10.

38 As well as Luke 10, Luke 17 describes a Samaritan who is the only one of the ten healed lepers who returns to thank Jesus.

5 Testing the Algorithm

This paper presents a generic NLP algorithm for detecting potential textual references. The algorithm itself is not particularly novel since it is in essence just the three steps of NLP. It was developed and tested during the research phase of a PhD in the field of biblical studies. The research looked at how the Pastoral Epistles (i.e. Titus and 1 & 2 Timothy) might have been influenced by the Septuagint and Jewish Pseudepigrapha. Implementing the algorithm as a computer program was not possible (nor necessary) in this context because a database of these texts was not available in the public domain. Since the semantic analysis (the most complicated aspect of NLP) needed to be manually performed, the lexical and syntax analysis was also simulated manually. This was possible because the algorithm is essentially very simple. The complicated parts of the process were the development of the syntax rules and the semantic analysis of the potential references.

In order to test the algorithm, a broad set of syntax rule definitions were developed that would seem to cover the different reference forms in Ancient Greek literature (see Section 2 above). The algorithm was initially used to detect potential references between the Pastoral Epistles and the Septuagint. The parameters of the algorithm were trained so that algorithm would detect all the potential references listed in the standard Greek editions of the New Testament (i.e. the UBS5 and the NA28) that the semantic analysis deemed to be meaningful. During this training, additional reference forms were added to the original set (including a definition of emphatic keywords). Other reference forms (such as multiple keywords) were taken out of the list because they were deemed not necessary for the Pastoral Epistles. The syntax rules were effective for the Pastoral Epistles. Future studies might test their effectiveness for other Ancient Greek texts.

Having trained these parameters to work for one set source texts (i.e. the Septuagint), the algorithm was then applied to a relatively new question, namely, "What are the potential references between the Pastoral Epistles and the Jewish Pseudepigrapha?". The algorithm was able to detect 36 potential references, which is substantially higher than the number detected by all previous studies (i.e. 12, of which only 6 were deemed as meaningful by the semantic analysis). The average verbal similarity of the algorithm's references (2.5 root words and 3.2 total words) was also higher than previous studies (2.2 root words and 2.8 total words). These results highlight the potential benefits of this approach.

6 The Benefits of the Algorithm

In recent years, several projects in the Digital Humanities have sought to introduce a level of automation to the search for textual references. In the context of these studies, this paper proposes a new algorithm that uses NLP. It is proposed that this algorithm be implemented as a web service so that can be used with the latest NLP technology, like Siri or Cortana, in order to perform new research. This algorithm presents three significant benefits for the study of the humanities.[39] Firstly, because it can be configured to detect potential references with low verbal similarity, it enables a systematic approach to the detection of allusions and influences. Secondly, it can search through large collections of source texts, even unfamiliar ones, so that more potential references can be considered. Then thirdly, this ability to perform large-scale searches means that metadata can also be collected, including which source text is used most frequently.

The primary benefit of this algorithm is its ability to be configured to detect a variety of reference forms (including those that have low verbal similarity) without overburdening the task of semantic analysis. This is particularly helpful for allusions, which can be signaled by a single keyword like Μελχισέδεκ ("Melchizedek"), as well as for influences, which might borrow only one or two words from their source text (or perhaps just synonyms of those words). As such, the detection of these references is frequently subjective and difficult to evaluate.[40] For example, the opening words of 1 Tim 1:15 (πιστὸς ὁ λόγος καὶ πάσης ἀποδοχῆς ἄξιος[41]) indicates that what follows (ὅτι Χριστὸς Ἰησοῦς ἦλθεν εἰς τὸν κόσμον ἁμαρτωλοὺς σῶσαι[42]) might contain a reference to another text,[43] but because there is no obvious quotation, it is difficult to determine which text this might be. However, by analyzing the frequency of different combinations of the words, such as how often ἔρχομαι ("I come") and σῴζω ("I save") are found together in the Septuagint, this algorithm can indicate which source text might have been the most influential.[44]

39 The first benefit applies only to this new algorithm. The remaining two benefits also apply to other methods of automation.

40 This subjectivity is highlighted by the observation that the NA28 lists as many as thirty-two potential references to the Septuagint in 1 Timothy, while the UBS5 has only twenty.

41 Eng. Trans. – "This word is faithful and worthy of all acceptance."

42 Eng. Trans. – "that Christ Jesus cam into the world to save sinners."

43 The UBS5 and NA28 both suggest a possible influence from the words of Jesus in Luke 19:10. The other possible influences are discussed in Towner, Philip H., *The Letters to Timothy and Titus*, USA: William B. Eerdmans Publishing Company, 2006, 145.

44 The idea of the messiah coming to save (ἔρχομαι and σῴζω, or their synonyms) is surprisingly rare in the Septuagint. Interestingly, one of these occurrences, Zech 9:9, is quoted

TABLE 6.2 Comparison of potential references from Titus to the Septuagint; ©BRETTGRAHAM

	Potential references	Average root words / reference	Average total words / reference
New Algorithm	38	2.5	3.2
UBS^5	7	2.3	2.9
NA^{28}	12	2.3	2.9

TABLE 6.3 Potential references from Titus to Septuagint – categorized by type; ©BRETTGRAHAM

	New algorithm	Those not in UBS^5 or NA^{28}
Quotations	8	5
Paraphrases	9	5
Single Keywords	1	1
Multiple Keywords	8	8
Structural Parallels	12	10
TOTAL	38	29

The efficiency of the new algorithm is demonstrated by analyzing the potential references from Titus to the Septuagint (see Table 6.2 above).

The new algorithm detected substantially more potential references compared to both the UBS5 and NA28 and yet these references also have greater verbal similarity, both in terms of matching root words and total matching words (i.e. including synonyms). The increase is partly due to the inclusion of more reference forms, with over 55% of the algorithm's references (i.e. 21 of 38) being keywords and structural parallels (as illustrated in Table 6.3 above). However, there were also five new paraphrases and five new quotations detected.

The additional references also make the total references more evenly distributed, both in terms of the places in Titus where they are found, as well as the locations of the relevant source texts in the Septuagint. This highlights that the algorithm functions equally well across all source texts, including those that are less familiar.

twice in the Gospels when Jesus makes his final entry into Jerusalem, suggesting that it was well known (and influential) within the Early Church.

This feature leads to the second major benefit of the algorithm for the study of humanities: namely, its ability to ask new questions. For example, the statement in 1 Tim 2:14, Ἀδὰμ οὐκ ἠπατήθη ("Adam was not deceived"), is usually understood by Biblical scholars as an allusion to Gen 3:13, where Eve explains, Ὁ ὄφις ἠπάτησέν με ("the serpent deceived me").[45] This connection comes from recognizing that this is the only verse in the Septuagint where the verb ἀπατάω ("I deceive") appears in the context of the noun Ἀδάμ ("Adam"). However, if the intertextual framework (i.e. the familiar source texts) of the author of 1 Tim 2 included the texts of Jewish Pseudepigrapha that were extant in the first century, then this same statement could instead be a rebuttal of Apoc. Sedr. 5:1 – ἠπατήθη [...] ὁ Ἀδάμ ("Adam was deceived"; c.f. LAE 16:5; Hist. Rech. 7:8; 3 Bar. 4:8; and Sib. Or. 1:39-40). Thus, by simply changing the set of source texts to search, the new algorithm can simulate a different intertextual framework and thereby provide insight into another possible context.

Finally, the third major advantage of automating the search for potential references is the ability to gather metadata. Although this is also possible for manual analysis, computers increase the scale by which it can be done. This then allows statistics to be collated, such as the number of times an author appears to refer to each source text, which in turn indicates which ones were most influential. Alternatively, by recording references to a particular source text across centuries (or domains), this data can highlight when (or where) that text had the greatest impact. This data can then be used in further analysis, such as tracking the usage of a text before and after certain major events, like wars and revivals, or even to study which texts have been the most influential in legal decisions. Consequently, the ability to record metadata provides a wealth of possibilities for more scholars to investigate further.

In summary, this new algorithm offers three significant benefits to the study of the humanities. The foremost of these is that it can detect any reference form, including those with low levels of verbal similarity. Furthermore, by varying the set of source texts to search, it can simulate different intertextual frameworks, thereby allowing it to highlight potential references in large databases of source texts, even those involving in less familiar texts. While doing so, metadata can also be collected that can in turn provide a better understanding of the way different source texts are used.

45 See, for example, Knight III, George W., *The Pastoral Epistles, New International Greek Testament Commentary*, Marshall, Howard, Gasque, W. Ward, eds., USA: William. B. Eerdmans Publishing Company, 1999, 143; Aland, Barbara, et al., eds., *Greek New Testament* (5th edition), Stuttgart: German Bible Society, 2014, 694; Aland, Barbara, et al., eds., *Novum Testamentum Graece: Nestle-Aland* (28th edition), Stuttgart: German Bible Society, 2012, 637.

References

Aland, Barbara, Aland, Kurt, Karavidopoulos, Johannes, Martini, Carlo M., Metzger, Bruce, eds., *Greek New Testament*, (5th edition), Stuttgart: German Bible Society, 2014.

Aland, Barbara, Aland, Kurt, Karavidopoulos, Johannes, Martini, Carlo M., Metzger, Bruce M., eds., *Nestle-Aland Novum Testamentum Graece*, (28th edition), Germany: Deutsche Bibelgesellschaft, 2012.

"Amazon Alexa". *Wikipedia*, May 18, 2017, <https://en.wikipedia.org/wiki/Amazon_Alexa>, accessed on 10.04.19.

Büchler, Marco, GeBner, Annette, Eckart, Thomas, Heyer, Gerhard, "Unsupervised Detection and Visualisation of Textual Reuse on Ancient Greek Texts", *Journal of the Chicago Colloquium on Digital Humanities and Computer Science 1/2*, June 16, 2010, <https://knowledge.uchicago.edu/record/133>, accessed on 10.04.19.

Charlesworth, James H, "The Pseudepigrapha as Biblical Exegesis", in: *Early Jewish and Christian Exegesis: Studies in Memory of William Hugh Brownlee*, Evans, Craig A., Stinespring, William F., eds., Atlanta: Scholars Press, 1987, 139-152.

Clough, Paul, Gaizauskas, Robert, Piao, Scott S.L., Wilks, Yorick, "METER: MEasuring TExt Reuse", *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL'02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, <http://eprints.whiterose.ac.uk/78530/>, accessed on 10.04.19.

"Cortana (software)", *Wikipedia*, May 31, 2017, <https://en.wikipedia.org/wiki/Cortana>, accessed on 10.04.19.

Crawford, Sidnie White, *Rewriting Scripture in Second Temple Times*, Grand Rapids: William. B. Eerdmans Publishing Company, 2008.

Dale, Robert, Moisl, Hermann, Somers, Harold, eds. *Handbook of Natural Language Processing*, (1st edition), New York: CRC Press, 2000.

Gamble, Harry Y., *Books and Readers in the Early Church: A History of Early Christian Texts*, New Haven: Yale University Press, 1995.

"Google Assistant", *Wikipedia*, May 17, 2017, <https://en.wikipedia.org/wiki/Google_As sistant>, accessed on 10.04.19.

Hartog, Paul, *Polycarp and the New Testament the Occasion, Rhetoric, Theme, and Unity of the Epistle to the Philippians and Its Allusions to New Testament Literature*, Tübingen: Mohr Siebeck, 2002.

Hays, Richard B., *Echoes of Scripture in the Letters of Paul*, New Haven: Yale University Press, 1989.

Hays, Richard B., *The Conversion of the Imagination: Paul as Interpreter of Israel's Scripture*, Grand Rapids, Mich.: William B. Eerdmans Publishing Company, 2005.

Van den Hoek, Annewies, *Clement of Alexandria and His Use of Philo in the Stromateis: An Early Christian Reshaping of a Jewish Model*, Leiden: E.J. Brill, 1988.

Hughes, Julie A., *Scriptural Allusions and Exegesis in the Hodayot*, Leiden: Brill, 2006.

Kittel, Bonnie Pedrotti, *The Hymns of Qumran*. Ann Arbor, Michigan: Scholars Press, 1981.

Knight III, George W., *The Pastoral Epistles, New International Greek Testament Com*, Marshall, Howard, Gasque, W. Ward, eds., USA: William. B. Eerdmans Publishing Company, 1999.

Lee, John, "A Computational Model of Text Reuse in Ancient Literary Texts", *Proceedings of the 45th Annual Meeting on Association for Computational Linguistics*, Prague, Czech, June 23-30, 2007, 472-479.

Louden, Bruce, *Homer's Odyssey and the Near East*, Cambridge: Cambridge University Press, 2011.

McLean, Bradley H., *Citations and Allusions to Jewish Scripture in Early Christian and Jewish Writings through 180 C. E,* Lewiston, NY: The Edwin Mellen Press, 1992.

Olsen, Mark, Horton, Russell, Roe, Glenn, "Something Borrowed: Sequence Alignment and the Identification of Similar Passages in Large Text Collections", *Digital Studies / Le Champ Numérique 2/1*, May 17, 2011, <https://www.digitalstudies.org/articles/10.16995/dscn.258/>, accessed on 10.04.19.

Perri, Carmela, "On Alluding", *Poetics 7*, 1978, 289-307.

Runia, David T., *Philo of Alexandria and the Timaeus of Plato*, Leiden: E.J. Brill, 1986.

"Siri", *Wikipedia*, May 18, 2017, <https://en.wikipedia.org/wiki/Siri>, accessed on 10.04.19.

Stanley, Christopher D., *Arguing With Scripture*, New York: T & T Clark, 2004.

Towner, Philip H., *The Letters to Timothy and Titus*, USA: William B. Eerdmans Publishing Company, 2006.