Qualitative Analysis of Semantic Language Models

Thibault Clérice, Matthew Munson

CHAPTER 5

Qualitative Analysis of Semantic Language Models

Thibault Clérice and Matthew Munson

1 Introduction

The task of automatically extracting semantic information from raw textual data is an increasingly important topic in computational linguistics and has begun to make its way into non-linguistic humanities research.[1] That this task has been accepted as an important one in computational linguistics is shown by its appearance in the standard text books and handbooks for computational linguistics such as Manning and Schuetze Foundations of Statistical Natural Language Processing[2] and Jurafsky and Martin Speech and Language Processing.[3] And according to the Association for Computational Linguistics Wiki,[4] there have been 25 published experiments which used the TOEFL (Test of English as a Foreign Language) standardized synonym questions to test the performance of algorithmic extraction of semantic information since 1997 with scores ranging from 20% to 100% accuracy.

The question addressed by this paper, however, is not whether semantic information can be automatically extracted from textual data. The studies listed in the preceding paragraph have already proven this. It is also not about trying to find the best algorithm to use to do this. Instead, this paper aims to make this widely used and accepted task more useful outside of purely linguistic studies by considering how one can qualitatively assess the results returned by such algorithms. That is, it aims to move the assessment of the results returned by semantic extraction algorithms closer to the actual hermeneutical tasks carried out in the, e.g., historical, cultural, or theological interpretation of texts. We believe that this critical projection of algorithmic results back onto the

1 Munson, Matthew, *Biblical Semantics Applying Digital Methods for Semantic Information Extraction to Current Problems in New Testament Studies*, Theologische Studien, Aachen: Shaker Verlag, 2017.

2 Manning, Chris, Schütze, Hinrich, *Foundations of Statistical Natural Language Processing*, Cambridge, MA: MIT Press, 1999.

3 Jurafsky, Daniel, Marin, James H., *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Second Edition, Prentice Hall Series in Artificial Intelligence, Upper Saddle River, NJ: Pearson Education, 2009.

4 <https://aclweb.org/aclwiki/TOEFL_Synonym_Questions_(State_of_the_art)>, accessed on 10.04.19.

hermeneutical tasks that stand at the core of humanistic research is largely a desideratum in the current computational climate. We hope that this paper can help to fill this hole in two ways. First, it will introduce an effective and yet easy-to-understand metric for parameter choice which we call Gap Score. Second, it will actually analyze three distinct sets of results produced by two different algorithmic processes to discover what type of information they return and, thus, for which types of hermeneutical tasks they may be useful. Throughout this paper, we will refer to the results produced by these algorithms as "language models" (or simply "models") since what these algorithms produce is a semantic model of the input language which can then help answer questions about the language's semantics. Our purpose in doing this is to demonstrate that the accuracy of an algorithm on a specific test, or even a range of tests, does not tell the user everything about that algorithm. We assert that there are cases in which an algorithm that might score lower on a certain standardized test may actually be better for certain hermeneutical tasks than a better scoring algorithm.

Much of the impetus for this study comes from the insights in Schnabel, et al. "Evaluation methods for unsupervised word embeddings", especially their assertion that an algorithm's performance on a standardized test does not reveal everything about that algorithm.[5] They demonstrate convincingly that the correct choice of an algorithm depends upon the type of task that it is expected to perform. They then go on to demonstrate that some algorithms are better at some tasks than other algorithms that are better at other tasks. In this study we suggest that one very effective way to determine whether an algorithm produces results that are useful for a certain task is to do a close reading of a portion of the results to determine whether these results will actually be valuable for the task at hand.

Another way that this Schnabel, et al., article is useful for the present study comes from the fact that the Gap Score metric we present here relies heavily in its conception on the "Coherence" task explained there.[6] In the Coherence task, three closely related words and one outlier are chosen from different language models. In their study, they then tested the results using crowdsourcing techniques, asking the crowdsourcers to choose the outsider and then measuring how often that outsider was the same as the one chosen by the algorithm. Gap Score presents a way to perform this task without crowdsourcing if one

5 Schnabel, Tobias, et al., "Evaluation Methods for Unsupervised Word Embeddings," in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, September 17-21, 2015, Lisbon, Portugal*, 2015, 298-307, <http://www.aclweb.org/anthology/D15-1036>, accessed on 10.04.19.

6 Schnabel et al., 302-303.

has an external categorization of semantically related words for a language or a specific corpus.

This study is broken down into two parts. The first part introduces the Gap Score metric and applies it to the results produced on the Greek Biblical corpus by the Word2Vec machine learning algorithm. In this study, we produced several language models based on different parameters using Word2Vec as implemented in the Python Gensim package and evaluated each of these using both simple distance measures and Gap Score. The second part of the study considers the most similar words according to the best scoring Word2Vec models and a different semantic-extraction algorithm, which is similar to that used by Bullinaria and Levy.[7] We chose these two algorithms because Word2Vec is widely considered to be one of the most effective algorithms for discovering word relationships and the algorithms that Bullinaria and Levy used produced the highest published accuracy on the TOEFL Synonym Questions task as reported by the ACL Wiki (see link above). We will analyze the patterns of these three models to discover what light the similarities and differences between these two lists shed on the type of semantic information returned by the two different algorithms.

2 Word2Vec and Gap Score

2.1 *Word2Vec*

The basic theory that underpins most methods for automatic extraction of semantic information is the distributional hypothesis. The most widely used explanation of this hypothesis is a pithy quote from British linguist John Rupert Firth, who wrote, "You shall know a word by the company it keeps!"[8] But two citations that explain the theory a bit better are from the American linguist Zellig Harris, who coined the term "distributional" to describe this phenomenon. In 1954 he wrote, "If we consider words or morphemes A and B to be more different in meaning than A and C, then we will often find that the distributions of A and B are more different than the distributions of A and C. In other

7 Bullinaria, John A., Levy, Joseph P., "Extracting Semantic Representations from Word Co-Occurrence Statistics: A Computational Study," 2007, <https://www.cs.bham.ac.uk/~jxb/PUBS/BRM.pdf>; Bullinaria, John A., Levy, Joseph P., "Extracting Semantic Representations from Word Co-Occurrence Statistics: Stop-Lists, Stemming and SVD," 2012, <http://www.cs.bham.ac.uk/~jxb/PUBS/BRM2.pdf>.

8 Firth, John Rupert, "A Synopsis of Linguistic Theory 1930-1955," in: Firth, John Rupert, *Studies in Linguistic Analysis*, Oxford: Blackwell, 1957, 11.

words, difference of meaning correlates with difference of distribution."[9] The most developed expression of this hypothesis came in a series of lectures that Harris did in 1986 in which he stated, "The most precise way of determining a word's meaning is by investigating the meanings of the words that occur along with that word."[10] Both the Word2Vec method presented here and the "Log-Likelihood" method, which we briefly explain in section 3 below, depend on Harris' distributional hypothesis to extract semantic representations of the words in a corpus.

The Word2Vec model is a shallow neural network model that was built by a team at Google headed by Tomas Mikolov in 2013[11] that has been used and studied very heavily since then. We will not undertake a technical, complex, or in-depth explanation of Word2Vec as we believe that this is beyond the scope of this paper. Instead we would refer the reader to the several articles published by Mikolov and his team[12] or any of the less technical explanations one can find in traditional publications[13] or on the internet.[14] Instead, the discussion here will focus on a basic, non-technical description of neural networks in general and Word2Vec's place within them.

A neural network is essentially a machine-learning method that has one or more hidden layers of "neurons" between the input layer and the output layer. The input layer, in the case of Word2Vec, is the textual material that we feed to it and the output layer is the result vectors that are produced. A neural network can "learn (progressively improve performance) to do tasks by considering

9 Harris, Zellig, "Distributional Structure," *Word* 10, no. 23, 1954, 156.

10 Harris, Zellig, "How Words Carry Meaning", Language and Information: The Bampton Lectures, Columbia University, 1986, <http://www.ircs.upenn.edu/zellig/3_2.mp3>, accessed on 10.04.19.

11 Mikolov, Tomas et al., "Distributed Representations of Words and Phrases and Their Compositionality," *CoRR* abs/1310.4546, 2013, <http://arxiv.org/abs/1310.4546>, accessed on 10.04.19.

12 Mikolov, Tomas et al., "Efficient Estimation of Word Representations in Vector Space," *CoRR* abs/1301.3781, 2013, <http://arxiv.org/abs/1301.3781>, accessed on 10.04.19; Mikolov, Tomas, Yih, Wen-tau, Zweig, Geoffrey, "Linguistic Regularities in Continuous Space Word Representations," in: *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, 2013, 746-751, <https://www.aclweb.org/anthology/N13-1090>, accessed on the 10.04.19; Mikolov et al.,"Distributed Representations."

13 Goldberg, Yoav, Levy, Omer, "Word2vec Explained: Deriving Mikolov et Al.'s Negative-Sampling Word Embedding Method," *CoRR* abs/1402.3722, 2014, <http://arxiv.org/abs/1402.3722>; Wikipedia, *Word2vec – Wikipedia, The Free Encyclopedia*, 2017, <https://en.wikipedia.org/w/index.php?title=Word2vec&oldid=785880094>, accessed on 10.14.19.

14 <https://youtu.be/D-ekE-Wlcds>, accessed 02-Feb-2018, <http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/>, accessed on 10.04.19.

examples."[15] In the case of this article, the "examples" that we provided to Word2Vec were the texts themselves and the training involved Word2Vec progressively testing sets of neurons to see how well these neurons could predict the contexts of the words in the corpus. So, for instance, if the phrase Ἰησοῦς Χριστός occurs frequently in our corpus, Word2Vec will try to find a set of transformations that will often predict Χριστός when it sees Ἰησοῦς, and vice versa.

It is helpful to imagine the neurons that sit between the input and output layers as neurons in the human brain. The neurons in our brain have heard enough of our own native language that when it receives the input of a certain sentence, say "Every day I drink apple ???", a certain set of neurons fires and produces, as output, the expected word represented by "???" in the sentence. A very likely result for the word to fill this context would be "juice". But if the person speaking the sentence finished it with the word "car", we would assume that they made a mistake and ask them whether they actually meant "juice". You could also picture these neurons as being related to certain concepts. So, for instance, there could be a "fruit" neuron that would be activated when it sees the word "apple" or "orange". And then there might be a "citrus" neuron that would be activated when it sees the word "orange" or "lemon". And these two neurons together would be able to tell you that "orange" is more similar to "lemon" than it is to "apple". Word2Vec tests during the training whether the corpus actually needs a neuron for fruit and one for citrus. If having these two neurons improves the results, then it will keep them. Then during the training process Word2Vec trains certain neurons to fire when given an input context so that the output word that is produced will match as closely as possible to the input texts that it has been given. And it tries to do this for *all* of the input contexts in the corpus at once!

Once the training process is finished, the results vectors are essentially the record of precisely which neurons fire and how strongly they fire for each of the words in the corpus.[16] So, in our fruit and citrus example above, it would record that the fruit neuron fires strongly for "apple", "orange", and "lemon", while the citrus neuron fires strongly only for "orange" and "lemon". The intuition then is that words that have similar neuron firing pattern vectors in the

15 Wikipedia, *Artificial Neural Network – Wikipedia, The Free Encyclopedia*, 2017, <https://en.wikipedia.org/w/index.php?title=Artificial_neural_network&oldid=787575153>, accessed on 10.04.19.

16 And the length of these vectors is determined by the size of the neural network, i.e., the number of neurons it has. So, for instance, if we have a corpus with a 1M word vocabulary and we use 1,000 neurons to describe its semantics, our results matrix is only 1M × 1K cells instead of a 1M × 1M that would be produced in, e.g., the Log-Likelihood method: 1/1000 the size.

results will have similar meanings. This means that one should be able to determine the similarity of two words by calculating the similarity of their results vectors using some similarity metric (typically cosine similarity). This whole explanation is a vast oversimplification of the actual process and, as with any such oversimplification, is not completely accurate in its description. For instance, we would not expect any of the neurons to have functions as well-defined as "fruit" or "citrus". Their functions and what causes each one to fire is actually much more complex and always dependent on the corpus that we give it. We believe, however, that this oversimplification is useful to understand what is happening during the training process of Word2Vec and, thus, to help to understand better the results that Word2Vec produces.

2.2 *Gap Score*

The Gap Score metric is our contribution to the evaluation of vector-space models for semantic domain extraction. It is based on the intuition that the difference (i.e., the "gap") between the mean similarity scores for a target word of the X most similar words as computed by a certain algorithm and the Y most similar words from an external testing set, i.e., the "in-domain" words (e.g., a list of words in a semantic domain), will be smaller than the difference between the mean similarity scores of that same target word of the X most similar words computed by an algorithm and one or more words (the "out-of-domain" words) that do not fall into the target word's external testing set. As noted above, we follow Schnabel, et al., in that we allow the algorithm to produce its own semantic category by taking the X most similar words to the target word. Then we compare candidates from externally produced categories to the algorithmically produced category to see how well the internal and external categories match each other.

Mathematically, the Gap Score metric is represented by the following equations:

$$WordScore(w, W) = \frac{\sum_{w_n \in W} \left(\left[\frac{\sum_{w_t \in T_{wn}} \mathrm{SIM}(w_t, w_n)}{|T_{wn}|} \right] - \max\left(\mathrm{SIM}(w, w_n), 0\right) \right)}{|W|} \quad (1)$$

where

- w represents a single word from a semantic domain
- W represents a set of words w is tested against
- T_{wn} the set of top X most similar words to wn according to some algorithm
- w_n represents each individual word from W that is tested

– w_t represents each individual word from T_{wn} to which w_n is compared
– SIM is the similarity metric that is used to compare the words with each other (e.g., Word2Vec)

$$DomainScore(W) = \frac{\sum_{w_n \in W} WordScore(w_n, W)}{|W|} \quad (2)$$

where *W* represents a set of words.

And the objective of this testing is to find a set of parameters *P* that results in maximizing $|DomainScore(W \cup O) - DomainScore(W)|$ where *W* represents a set of words from a semantic domain and *O* represents a set of words from a disconnected semantic domain. This result in a Gap Score is positive if the in-domain words are more similar to each other but is negative if the out-of-domain words fit better. Also the distance of the Gap Score from 0 reflects the difference between the in-domain and out-of-domain words. If the in-domain words are significantly more similar to the X most similar words, then the score will be significantly above 0, whereas if the out-of-domain words are significantly more similar to the same X words, then the score will be significantly below 0. The code to carry out the gapscore algorithm was written in Python and, along with thorough documentation on its use, is openly available on Github at <https://github.com/hipster-philology/param-bias>.

2.3 *Evaluation Procedure*

Once one has one or more vector-space language models of the corpus under investigation, the next task is to evaluate how these models performed. As semantic categories, we have used the semantic sub-domains from the Louw-Nida lexicon, which can be found online.[17] This online data represents the domains and sub-domains of the printed edition of this lexicon[18] and is based on the theoretical work done by the authors of the lexicon.[19] When we say "sub-domains", we mean the collections of words represented by, e.g., domain "1A Universe, Creation" as opposed to using the whole primary domain, e.g., "1 Geographical Objects and Features". And we have only included sub-domains that have at least 10 words whose primary meaning belongs to

17 <http://www.laparola.net/greco/louwnida.php>, accessed on 10.04.19.

18 Louw, Johannes P., Nida, Eugene A., *Greek-English Lexicon of the New Testament: Based on Semantic Domains*, Second Edition, 2 vols., New York: United Bible Societies, 1989.

19 Nida, Eugene A., *Componential Analysis of Meaning*, The Hague: Mouton, 1975; Nida, Eugene A., Louw, Johannes P., Smith, Rondal B., "Semantic Domains and Componential Analysis of Meaning," in: *Current Issues in Linguistic Theory*, ed. Roger William Cole, Bloomington: Indiana University Press, 1977, 139-167.

that sub-domain. In the online version of the Louw-Nida lexicon, the primary meaning of a word is represented either by a "Gloss" that has no letter before it (for words with only a single gloss) or that is preceded by the letter "a". We have also not included words that are only represented by a phrase in a certain domain. Take, for instance, the domain "4A Animals".[20] In this domain, the first two words, ζωή and ψυχή appear only in the phrase ψυχὴ ζωῆς (living creature) and thus are excluded from the words in our cleaned sub-domain. Also, the word υἱός is excluded from this domain since the prefixed "d" means that this is actually the quaternary instead of the primary meaning. And then we selected only the first 10 words in each sub-domain since, according to the Louw-Nida organizational scheme of placing the words that are most generally related to the sub-domain first, these should be the words that best represent the sub-domain as a whole. These filters resulted in a cleaned sub-domain 4A that contains only the ten words ζῷον, θηρίον, τετράπους, θρέμμα, κτῆνος, ὑποζύγιον, ἀγέλη, ἀλώπηξ, λύκος, and ἄρκος.[21]Once we had cleaned all Louw-Nida sub-domains, we were left with 56 sub-domains that had at least 10 members. We then randomly produced 100 sub-domain pairs for testing and then, for each of these pairs, we produced a list of words that was made up of 3 words from the first domain and a single word from the second domain.[22] We chose to use sets of 3 in-domain words and 1 out-of-domain word in order to mirror the Coherence test in Schnabel, et al.[23] Then we evaluated these lists of words in two ways. First, we allowed Gensim's `doesnt_match` function on its Word2Vec model[24] to pick the single word in this list that fits worst. This function calculates the mean similarity for all of the given words with all of the other given words and chooses the one word that is least similar to the other words. So, for instance, if we gave the `doesnt_match` function the list of words "breakfast cereal dinner lunch", we would expect it to return the word "cereal" as the non-matching word.[25] For our tests, if this word was the out-of-domain word, then that whole list of words received a score of 1. If it was actually one of the in-domain words, the list received a score of 0. We then also computed the Gap

20 <http://www.laparola.net/greco/louwnida.php?sezmag=4&sez1=1&sez2=37>, accessed on 10.04.19.

21 We also excluded the domains "89 Relations", "90 Case", "91 Discourse Markers", "92 Discourse Referentials", and "93 Names of Persons and Places" as domains whose primary relating factor is syntactic rather than semantic.

22 This list of test sets can be found in the Appendix.

23 Schnabel et al., "Evaluation Methods," 302-303.

24 gensim.models.keyedvectors.KeyedVectors#doesnt`match

25 <https://radimrehurek.com/gensim/models/keyedvectors.html#gensim.models.keyedvectors. KeyedVectors.doesnt_match>, accessed on 10.04.19.

Score for each list of words, using the top 5 most similar words as computed by Word2Vec for each word.

In our tests with Word2Vec, we manipulated four different parameters: the size of the context window, the text chunk sizes we used for input, the dimensionality of the resulting feature vectors, and whether we started the training process with a pre-trained model or not. We will explain these parameters in order. The size of the context window determined how many words to the left and to the right of the target word would be counted as valid co-occurrents. As explained above, Word2Vec depends on word co-occurrence counts for its calculations. So if we chose a window size of 5, all words within 5 words to the left and 5 words to the right of the target word would be counted as co-occurrents. The premise behind manipulating this parameter is that co-occurrents that tend to be semantically important to the target word will tend to occur closer to that word. But it is unclear precisely where the cutoff in a corpus comes where increasing the window size will result in an increase in random noise as opposed to an increase in semantic information. A higher performance for a smaller window size would lead to the conclusion that semantic information tends to be tightly focused within a corpus, e.g., with short, to-the-point sentences. While a better performance for a larger context window would suggest larger distances between semantically related items within the corpus, e.g., long, complex sentences. We tested window sizes of 5 words and 10 words.

The text chunk sizes determined how large the input text chunks were. We tested as input texts single biblical verses, single chapters, single books (e.g., Genesis or the Gospel of Matthew), and the whole Septuagint and New Testament as a single large text. In conjunction with the window size above, the text chunk size acted as a limit on the words that would be counted as semantically important. No matter what the window size used, the counting of co-occurrents could never extend beyond the boundary of the text chunks we used. So if we used the verse as the chunk size, all of the words within that verse would be considered co-occurrents with the target word if they fell within the window size. But no words from the next verse could possibly be chosen simply because they were not considered to be part of the text that we were testing. The thinking behind the manipulation of this parameter is similar to that for context window size above except instead of testing the relationships of single words to each other we were more testing how chunks of text were related semantically. So if, e.g., the chunk size of verses performed the best, that would mean that semantically related ideas are most concentrated on the level of the verses as opposed to the level of the chapter or the book. So if the performance would decrease for the larger text chunks, such as the chapter, that suggests

that, as above, expanding to this larger chunk adds more noise to the model than it adds information.

The size of the resulting feature vectors determined how many values the vectors for each word in the vocabulary contain. As explained above, Word2Vec learns the most salient features (the neurons) of a corpus by making repeated training runs over the corpus using different feature sets. Once the training is over, Word2Vec then chooses the set of neurons that does the best job in predicting word occurrence given a certain verbal context. By manipulating the size of the feature vectors, we were exploring how many neurons best described the corpus at hand. We tested vector lengths of 30, 50, 80, 100, and 200 neurons.

Finally, the parameter of starting with a pre-trained model or not meant that we either trained a brand new model based only on the biblical text chunks or we started with a model that had been trained on another, larger corpus and changed that model based on the new information in the biblical text chunks. If we started from scratch, the model that results would be based only on the biblical text and thus would theoretically represent a purely biblical Greek language model whereas starting with a model trained on a general Greek corpus would produce a more mixed model. The primary question we wished to answer by manipulating this parameter is whether there is enough data in the biblical corpus itself to produce a useful language model or not. So if the pre-trained models performed better, that suggests one of two things. Either there is not enough data in the biblical corpus to produce a good model OR that the training data that we are using (the Louw-Nida Lexicon) is based to a large extent on general Greek evidence as opposed to purely biblical evidence. The Louw-Nida Lexicon uses what they call "extratextual contexts",[26] i.e., evidence from outside of the biblical corpus, to assist in its definitions and its categorization because, as they assert, "the Greek of the New Testament should not be regarded as a distinct form of Greek, but rather as typical Hellenistic Greek."[27] The extent to which they have used such evidence, however, is difficult to measure. This parameter will then, at least in part, help us to see how prevalent non-biblical semantics are for Louw-Nida.

2.4 *Discussion of Results*

Table 5.1 below shows the top ten highest scoring parameter sets ordered by the mean Gap Score for all of the 100 input word sets. All of the table headings should be self-explanatory except perhaps "Size", which represents the number

26 Louw, Nida, *Greek-English Lexicon of the New Testament: Based on Semantic Domains*, xvi.
27 Louw, Nida, xvi.

TABLE 5.1 Top 10 best performing language models, NT and LXX: Mean gap score; ©CLERICEMUNSON

Text chunk size	Pre-trained?	Context window size	Size	Gap score correct	Gensim correct	Average gap score
Verses	Yes	5 Words	30	82	56	0.1428
Verses	No	5 Words	30	81	60	0.1352
Chapters	Yes	5 Words	30	78	63	0.1281
Verses	No	10 Words	30	78	57	0.1258
Chapters	No	5 Words	30	80	60	0.1250
Books	Yes	5 Words	30	75	54	0.1174
Books	No	5 Words	30	76	56	0.1134
Verses	Yes	10 Words	30	75	49	0.1113
Verses	No	5 Words	50	78	57	0.1083
Full Bible	Yes	5 Words	30	74	54	0.1081

of neurons in the result vectors, and "Gap Score Correct" and "Gensim Correct". "Gap Score Correct" measures how many times the Gap Score for a set of test words was positive, meaning that the in-domain words are, on average, closer to each other than they are to the out-of-domain word. "Gensim Correct" is how many times the Gensim `doesnt_match` function chose the correct out-of-domain word. These scores both have a possible maximum of 100, so "82" would mean that Gap Score correctly categorized 82 out of 100 test sets. Also note that for this paper, we used the Continuous Bag of Words (CBOW) method for Word2Vec, which is the default in Gensim.

For reasons of space, we will restrict the discussion here to the four test parameters listed in section 2.3 above. First notice that the most important parameter appears to be the size of the neural network. Nine out of the top ten results came from the smallest network of only 30 neurons. While this may seem surprising at first, we explain it as resulting from the thematically focused nature of the corpus. Both the Septuagint and the New Testament deal primarily with God's relationship with Israel and thus it requires fewer neurons to describe than a general English language corpus would require. The next most important parameter is the context window size. Eight of the top ten results had a context window of only 5 words as opposed to 10 words. This means that, in our corpus, the semantically important words tend to concentrate themselves within 5 words of the target word. Adding word numbers 6 to 10 to these calculations tends to add information that is not as closely related to the semantics of a word as the first 5 words are.

TABLE 5.2 Top 10 best performing language models, NT and LXX: Gensim; ©CLERICEMUNSON

Text chunk size	Pre-trained?	Context window size	Size	Gap score correct	Gensim correct	Average gap score
Chapters	Yes	5 Words	30	78	63	0.1281
Verses	No	5 Words	30	81	60	0.1352
Chapters	No	5 Words	30	80	60	0.1250
Full Bible	Yes	10 Words	30	73	59	0.0882
Chapters	No	5 Words	50	74	58	0.0829
Verses	No	10 Words	30	78	57	0.1258
Verses	No	5 Words	50	78	57	0.1083
Verses	Yes	5 Words	50	76	57	0.0978
Verses	Yes	5 Words	30	82	56	0.1428
Books	No	5 Words	30	76	56	0.1134

The next most important parameter is the text chunk size, with 5 of the top 10 being Verses, 2 each being Chapters or Books, and the Full Bible appearing at number 10. This suggests, as did the small context window size, that semantic information related to the words in a biblical verse tends to be more concentrated within that verse. And, finally, the choice starting with a pre-trained model or not appears to have very little effect on the results, with 5 of the top 10 having used a pre-trained model and 5 not using one.

Those were the results ordered according to Gap Score Average. Table 5.2 represents the top 10 best performing parameter combinations ordered according to Gensim's ability to correctly identify the outlier word.

In this table, we see that the most important parameter for Gensim appears to be the context window size. Eight of the top ten used 5-word context windows, just as we saw above in the Gap Score results. The next most important was the size of the neural network, with 7 of 10 using a 30-neuron network and the other 3 a 50-neuron network. The next most important was the text chunk size, with 5 having used Verses, 3 having used Chapters, and then 1 each having used Books or the Full Bible. And, finally, the least important was whether a pre-trained corpus was used. Four of the top ten used a pre-trained corpus while 6 did not.

The next two tables are organized the same way as the two tables above but, instead of using the whole Old and New Testament to train their language models, these are based on models trained using just the New Testament. We include these here for two reasons. First, we wish to discover whether there are

TABLE 5.3 Top 10 best performing language models, NT only: Mean gap score; ©CLERICEMUNSON

Text Chunk size	Pre-trained?	Context window size	size	Gap score correct	Gensim correct	Average gap score
Verses	No	10 Words	30	65	45	0.0715
Verses	Yes	5 Words	30	71	43	0.0688
Books	Yes	5 Words	30	61	43	0.0635
Verses	No	5 Words	30	61	47	0.0625
Chapters	Yes	5 Words	30	63	44	0.0617
Books	No	10 Words	30	61	36	0.0592
Verses	Yes	10 Words	30	68	40	0.0574
Chapters	No	10 Words	30	62	35	0.0509
Verses	Yes	5 Words	50	67	46	0.0488
Books	No	5 Words	30	59	42	0.0467

any differences in the best parameters based on corpus size. The New Testament is approximately one-fifth the size of the Septuagint and, thus, it could require different parameters to produce the best model. The second reason is that we need data on the best models for only the New Testament so that we can more easily compare the results in section 3 below. Table 5.3 represents the top ten according to Average Gap Score and Table 5.4 according to Gensim's `doesnt_match` function.These tables show the same preference for smaller input text chunks as the previous two tables, with the second, Gensim table actually having 7 of the top ten relying on the verse-level chunks. They also both show no preference for pre-trained data, with the Gap Score table having 5 pre-trained and 5 not pre-trained and the Gensim table with 4 and 6, respectively. This is perhaps a bit surprising since we might expect that a corpus as small as the New Testament (about 130,000 words) might benefit from a model that has already been pre-trained for general Greek. However, the results suggest that one can get just as good a language model without such pre-training. The Gap Score table still shows a marked preference for fewer neurons, with 9 out of the 10 having only 30. The Gensim table, however, prefers larger networks, with only 5 of the ten having 30 neurons, 3 having 50, and 2 having 80. This suggests that the `doesnt_match` function requires a more complex representation of the corpus in order to produce good results. Finally, the Gap Score results show more preference for the larger, 10-word context window than did the previous two tables, 4 of 10 depending on this window size. The

TABLE 5.4 Top 10 best performing language models, NT only: Gensim; ©CLERICEMUNSON

Text size	ChunkPre-trained?	Context window size	size	Gap score correct	gensim correct	Average gap score
Verses	No	5 Words	30	61	47	0.0625
Chapters	No	5 Words	50	66	47	0.0422
Verses	No	10 Words	80	59	47	0.0341
Verses	Yes	5 Words	50	67	46	0.0488
Verses	No	10 Words	30	65	45	0.0715
Verses	No	10 Words	50	63	45	0.0429
Verses	No	5 Words	80	64	45	0.0379
Chapters	Yes	5 Words	30	63	44	0.0617
Verses	Yes	5 Words	30	71	43	0.0688
Books	Yes	5 Words	30	61	43	0.0635

Gensim results also showed a slightly higher preference than the previous two, with 3 of 10, but still less so than Gap Score. This last observation about the preference for larger context windows for the New Testament probably comes from the corpus size of the New Testament. The larger context window collects more information for every word and thus makes up for the lack of evidence coming from the number of words in the corpus.

We should also point out that both evaluation metrics tended to score lower on the New Testament than on the combined Old and New Testaments, with the top number of correct predictions from Gap Score for the combined corpus being 82/100 and for the New Testament only 71/100. Gensim showed a similar pattern with 63/100 on the combined corpus and 47/100 on the New Testament alone. All of these scores, however, are significantly better than chance, which would result in a score of 25/100. So there is useful semantic information being captured for both corpora, which we will examine in more detail below.

This brief analysis of the top results has shown that both Gap Score and Gensim tend to prefer the same parameters for the full biblical corpus, i.e., a small neural network (30 neurons) with a small context window (5 words) and small chunks of text (verses). And pre-training on a general Greek corpus does not appear to affect performance at all. The number of neurons and the size of the context window tended to increase when we trained on only the New Testament, though the preference for the verse-sized chunks of text remained constant. We will perform a more in-depth comparison of the results of these

two evaluation metrics below when we actually compare the lists of the top 20 most similar words for the top performing parameter combinations for these two metrics, as well as the results from a different semantic extraction method that is based more closely on the method used by Bullinaria and Levy and that will be described in more detail below.

3 Semantic Information Extraction

The purpose of this part of the paper is to actually go in-depth into the results produced by three different language models for the extraction of semantic information from the Greek biblical corpus. The first two language models were discussed above and both were produced by Word2Vec, one being the top scoring model according to Gap Score and the other the top-scoring model according to Gensim's `doesnt_match` function. The third model was produced using a different method for semantic information extraction, though one that is still based on the distributional hypothesis, and thus word co-occurrences, for its results.

First we will briefly describe this differing method, which we will call the "Log-Likelihood" method, based on the hypothesis testing algorithm that sits at its heart. A fuller description can be found in Munson's dissertation[28] and in the 2007 article from Bullinaria and Levy.[29] This method is a simpler one than Word2Vec in that it simply counts the co-occurrents for each word in the corpus, then measures the statistical significance of these co-occurrence values using Dunning's Log-Likelihood ratio,[30] and then compares these resulting statistical significance vectors using the cosine similarity algorithm. The step of calculating statistical significance using the Log-Likelihood ratio is important to normalize the data for high and low occurrence words. If we did not do this step, the top co-occurrent for every word in the Greek New Testament would be ὁ, since this is the most frequent word in the corpus. By implementing a significance measure, this method is able to correct somewhat for extremely frequent and extremely infrequent words.

One major downside of the Log-Likelihood method is that the resulting matrices are extremely large, being N × N squares, where N is the size of the vocabulary in the corpus. So if you have an imaginary corpus that has a vocabulary

28 Munson, *Biblical Semantics*, 5-33.

29 Bullinaria, Levy, "Extracting Semantic Representations from Word Co-Occurrence Statistics: A Computational Study."

30 Dunning, Ted, "Accurate Methods for the Statistics of Surprise and Coincidence," *Computational Linguistics 19*, 1993, 61-74.

of 1M (1 million) words, the resulting matrix would be 1M × 1M, or 1 trillion, cells. Such a matrix, if it were filled with 64-bit floating point numbers in every cell, would take up 8TB of space, either in memory or on disk, making them very difficult to work with. Whereas a Word2Vec matrix that is 1M × 1K cells would only take up 8GB of space and, thus, could be handled easily by a modern computer.

Munson, in his dissertation, carried out extensive parameterization of this Log-Likelihood method and determined that the context window that best predicted the Louw-Nida semantic sub-domains was a weighted window of 12 words left and right. The term "weighted" here simply means that words that co-occurred closer to the target word were given more weight than those that occurred farther from the target word. Notice that this window is larger than the optimal window shown in our tests of Word2Vec above, which tended to prefer a 5-word window. It is also interesting to note that while the Log-Likelihood method performs better with a weighted context window, the Continuous Bag of Words algorithm used to produce the language models for Word2Vec actually uses an unweighted context window, i.e., weighting every word that co-occurs within the context window the same. Also, the text chunk size used to produce the language model for the Log-Likelihood method was the biblical book as opposed to the smaller biblical verse that was preferred by Word2Vec. And, finally, we should note here that for this study we ran the language model produced using these parameters by the Log-Likelihood method through Gap Score in order to compare it with the other two methods.[31] According to Gap Score, it was able to select the correct out-of-domain word 47 times out of 100. This was significantly worse than the performance shown in Table 5.3.[32]

But now we would like to move on to the comparison of the results from these three language models. To do this, we have chosen to focus on a single word from the New Testament, δαιμόνιον, which is typically translated as "demon" in English. We have chosen this word for several reasons. First, it is an interesting word that holds an important, though not central place in the New Testament. It occurs fairly frequently, though not too often (63 occurrences). And it has a single, well understood meaning. We will start with the table of 20 most similar words based on the Log-Likelihood model. This table, as well as Table 5.6 and Table 5.7, are sorted according to the word's similarity with δαιμόνιον as calculated by the appropriate algorithm. The glosses that we are

31 Note, however, that the Gap Score method was not the method used to assess the result in Munson's dissertation. Munson, *Biblical Semantics*, 15-17.

32 Note that we have no basis for comparison of this language model with the Gensim's doesnt`match function since that function requires a Gensim Word2Vec language model to work.

TABLE 5.5 Top 20 most similar words to δαιμόνιον: Log-likelihood model; ©CLERICEMUNSON

1	βεελζεβούλ (7)	Beelzebub	11	κριτής (19)	Judge
2	ἐκβάλλω (81)	throw out	12	ἑλληνίς (2)	Greek
3	ἄρχων (37)	ruler	13	φθάνω (7)	come to
4	ἔξω (62)	outside	14	ὀλιγοπιστία (1)	poverty of faith
5	κωφός (14)	mute	15	τράπεζα (15)	Table
6	διαβλέπω (3)	see clearly	16	δαιμονίζομαι (13)	to be demon possessed
7	συροφοινίκισσα (1)	Syrophoenician	17	νόσος (11)	Sickness
8	θανάσιμον (1)	deadly	18	σός (25)	Your
9	ἔννυχα (1)	at night	19	δοκός (6)	beam (of wood)
10	θεραπεύω (43)	heal	20	βασιλεία (162)	Reign

using for the Greek word are based on the primary gloss that is given for each word.[33] The number in parentheses after each Greek word is the number of occurrences that word has in the New Testament.

The group of words in Table 5.5 is very clearly about demons, demon possession, and exorcism. βεελζεβούλ, ἄρχων, and βασιλεία all refer to the kingdom and rulers of demons while ἐκβάλλω, θεραπεύω, and δαιμονίζομαι all refer to demon possession and exorcism. συροφοινίκισσα and ἑλληνίς refer to the specific exorcism story in Mark 7:24-30 while τράπεζα is used in this same story both here and in Matthew 15:21-28. Almost all of the other words in this list refer to the miracles that Jesus performed in the Gospels: κωφός and νόσος refer to the sickness that is healed, διαβλέπω and (again) θεραπεύω refer to the miraculous healing, and ἔξω and φθάνω all set the scene for the miracle (φθάνω refers to the people coming to Jesus). And, finally, κριτής, ὀλιγοπιστία and θανάσιμον are on this list because they occur in the context of miracle stories or exorcisms in general. The first is used when speaking of exorcism in Matthew 12:27 and Luke 11:19, the second in the exorcism story at Matthew 17:20, and the third is used in Mark 16:18 in a verse that mentions miracles that Jesus' disciples will do. δοκός appears on this list because in all of its occurrences (Matthew 7:3-5 and Luke 6:41-42) it co-occurs with ἐκβάλλω, a word that is closely related to δαιμόνιον.[34]

The relationship of the words σός and ἔννυχα with δαιμόνιον is unclear. The former could show up because it appears with the word ἐκβάλλω in Matthew

33 <http://www.laparola.net/greco/louwnida.php>, accessed on 10.04.19.

34 See Munson, *Biblical Semantics*, 41-42, for deeper analysis of the related case of δοκός and δαιμόνιον.

TABLE 5.6 Top 20 most similar words to δαιμόνιον: Gap score model; ©CLERICEMUNSON

1	λεπρός (9)	leper	11	φραγέλλιον (1)	Whip
2	ἀμφότεροι (14)	both	12	λιβανωτός (2)	Censer
3	τόξον (1)	bow	13	χρεία (49)	what is needed
4	ἐκμυκτηρίζω (2)	ridicule	14	Ἰεριχώ (7)	Jericho
5	δαίμων (1)	demon	15	εὐθύς (59)	straight, immediately
6	χωλός (14)	lame	16	ὡσαννά (6)	Hosanna
7	κακῶς (16)	evil	17	Ἰάϊρος (2)	Jairus
8	νεανίσκος (10)	young man	18	ὀρθῶς (4)	correct(ly)
9	νομή (2)	pasture	19	ἀνάχυσις (1)	Excessive
10	τρίτον (1)	third part	20	γενετή (1)	Birth

7:3 and three times with ἐκβάλλω and δαιμόνιον in Matthew 7:22, though it is questionable whether only 4 out of the 25 occurrences of this word should so powerfully affect its semantic vector. The latter, however, would require more analysis to detect its relationship.

This list demonstrates, as shown more fully in Munson's dissertation,[35] that the Gospels set demons firmly within the context first of demon possession and exorcism and more generally into the context of Jesus' miracle stories in general. As Munson asserts, the role of demons in the New Testament is not so much as evil otherworldly beings but more so as a foil to demonstrate Jesus' power as a wonder worker. And this is the focus of the semantics that this semantic extraction method captures. Now we will consider the top 20 most similar words for the best New Testament Word2Vec model according to Gap Score (in Table 5.3: text chunks are verses, not pre-trained, 10-word context window, using 30 neurons).

In Table 5.6, it is interesting to notice the number of times each of these words occurs. This list of words has an average occurrence of 10.1 times in the New Testament and there are only two words, χρεία and εὐθύς, that occur more than 20 times. This is in contrast to the number of occurrences in Table 5.5, where the average number of occurrences is 25.55 and there are 5 words that occur more than 20 times. Though this is only a small sample size, looking only at the single word δαιμόνιον, it is interesting to consider perhaps that Word2Vec, or at least the best Word2Vec model according to Gap Score, might prefer less frequent words to the Log-Likelihood model enumerated in Table 5.5. We

35 Munson, Biblical Semantics, 40-44.

will wait until after we have analyzed the words in this table and Table 5.7 to comment further on this.

While this list of words may, at first glance, seem more random than that in Table 5.5.5, we actually see several of the same themes in this table as we saw there. First, the obvious words: δαίμων and κακῶς are related to δαιμόνιον in that the former is another word that refers to the same entity while the latter refers to their nature. We also see two words related to sickness, and thus probably to Jesus' miracles, in λεπρός and χωλός. But if we look closely at the contexts in which the other words appear on this list, we actually see that many of them actually appear in stories about Jesus' miracles, both demon exorcism and healing miracles. First there is εὐθύς. Of the 59 times that this word occurs in the New Testament, 42 of them are in the Gospel of Mark. And of these 42, it appears 17 times in the context of a miracle story[36] with a typical usage describing the immediacy of the healing (1:42, 2:12, 5:29, 5:30, 5:42 (2x), and 10:52). So εὐθύς is closely related to miracle stories. But it is also occur three times independently of any miracle story along with the word πνεῦμα (spirit), which is also the word used in the phrase πνεῦμα ἀκάθαρτον ("unclean spirit", e.g., Mark 3:30) to describe demons. So there appear to be two different usages of εὐθύς that tend to bring it into the distributional semantic space of δαιμόνιον: miracles and spirit.

Other words appearing often in miracle stories are ἀμφότεροι, which appears in a demon possession story in Acts 19:16, while νεανίσκος (Luke 7:14 and Acts 20:12), Ἰεριχώ (Matthew 20:29, Luke 10:35, Mark 10:46), Ἰάϊρος (Mark 5:22), ὀρθῶς (Mark 7:35), and γενετή (John 9:1) all appear in the context of healing miracles. And since all of these words occur fairly infrequently (the most frequent word is ἀμφότεροι, which occurs only 14 times), these occurrences within miracle stories carry a lot of weight in determining the semantics of these words. So, in the New Testament, all of these words have close verbal (though not necessarily semantic) relationships with miracle stories and are thus considered to be similar to demons because demons are also closely related to miracle stories. We could also perhaps include νομή in this list of miracle words since in one of its two occurrences (2 Timothy 2:17) it is used next to the word γάγγραινα, which names a certain class of diseases.

And then we see three words that appear to show up on this list because they co-occur with words that tend to co-occur with δαιμόνιον: τρίτον, φραγέλλιον, and ὡσαννά. The first two words occur with ἐκβάλλω (in Luke 20:12 and John 2:15, respectively), which is the word used in the New Testament for exorcising demons. And ὡσαννά because it is used in Mark 11:9, Matthew 21:9, and

36 1:30, 1:42, 1:43, 2:8, 2:12, 3:6, 5:2, 5:29, 5:30, 5:42 (2x), 6:54, 7:25, 9:15, 9:20, and 10:52.

TABLE 5.7 Top 20 most similar words to δαιμόνιον: Gensim model; ©CLERICEMUNSON

1	ἀμφότεροι (14)	both	11	εὐθύς (59)	straight, immediately
2	λεπρός (9)	leper	12	φραγέλλιον (1)	Whip
3	ἄλαλος (3)	mute	13	δαιμονίζομαι (13)	to be demon possessed
4	τόξον (1)	bow	14	Ἰάϊρος (2)	Jairus
5	ἐργασία (6)	behavior	15	ἱμάτιον (60)	Clothing
6	ὀρθῶς (4)	correct(ly)	16	ἕτερος (97)	Different
7	κωφός (14)	mute	17	ἡμιθανής (1)	half dead
8	βλασφημία (18)	reviling	18	μαλακός (4)	Soft
9	παιδίον (52)	child	19	ὅμως (3)	Although
10	χωλός (14)	lame	20	ἀγράμματος (1)	Uneducated

Matthew 21:15 along with the verb κράζω and in John 12:13 with the verb κραυγάζω, both of which mean "to cry out" and both of which are used to denote the action of demons in Mark 5:5, Mark 9:26, and Luke 9:39 (for κράζω) and in Luke 4:41 (for κραυγάζω). So these three words are included on this list because they co-occur with words that also co-occur with δαιμόνιον and, thus, according to the distributional hypothesis, share some semantic relationship with δαιμόνιον.

With the other 5 words, τόξον, ἐκμυκτηρίζω, λιβανωτός, χρεία, and ἀνάχυσις, we could find no discernible pattern as to why these words might be closely related to δαιμόνιον, though it is interesting to note that τόξον and λιβανωτός occur only in Revelation and both in the context of the action of heavenly being in relation to the seven seals (Revelation 6:2 and Revelation 8:3 and 8:5, respectively). With the words ἐκμυκτηρίζω, which is used in Luke 16:14 and 23:35 to describe people who are ridiculing Jesus, and ἀνάχυσις, used in 1 Peter 4:4 in conjunction with blasphemy, we might tentatively suggest that this group of four words might have to do with sin and judgment. But we think that this is far too tenuous a connection to really assert it at this point.

Table 5.7 shows the top 20 most similar words for the best scoring model according to Gensim's `doesnt_match` function: verse-sized text chunks, not pre-trained, 5-word context window, and 30 neurons. In this list of words, there are 9 that also show up in Table 5.6, and we will allow the explanation above to relate to the words in this table as well. We also see three words here that can be readily categorized according to the categories already mentioned in the two tables above: δαιμονίζομαι for demon possession and ἄλαλος and κωφός as sicknesses that Jesus heals. Then we have several words that occur regularly in miracle stories: ἐργασία appears in an exorcism story in Acts 16:16

and 16:19, παιδίον is often the object of a healing miracle (Mark 5:39, 5:40 (x2), 5:41, 9:24; Luke 7:28, 7:30; John 4:49), ἱμάτιον is used quite often as the object through which Jesus' power is channeled for healing (Matthew 9:20, 9:21, 14:36; Mark 5:27, 5:28, 5:30, 6:56) and it is mentioned in the miracle stories at Mark 10:50, Luke 8:27, Luke 8:44, and Acts 9:39. Then we see βλασφημία and ὅμως, which probably appear because they are both used with words that are closely related to δαιμόνιον: the former co-occurring with πνεῦμα in Matthew 12:31 and Mark 3:28 and the latter with ἄρχων in John 12:42. So the relationship of these two words with δαιμόνιον-related words brings them closer to δαιμόνιον. Then we have two words that co-occur with miracle-related words: ἡμιθανής in its only occurrence appears along with Ἰεριχώ which, as we saw in the explanation of Table 5.6 above, is closely related to miracle stories, and ἀγράμματος, which occurs in Acts 4:13 along with the verb θαυμάζω (to be amazed), a word which is regularly used to describe the amazement of the witnesses to a healing miracle (Matthew 8:10, 9:33, 15:31; Mark 5:20; Luke 7:9, 9:43, 11:14). So these words are related to δαιμόνιον because they all are related to miracles.

Of the last two unexplained words on this list, μαλακός has an even more tenuous connection to δαιμόνιον than the previous four words. Three of the four occurrences of μαλακός come in Jesus' description of John the Baptist as one who does not wear "soft" clothes. And John the Baptist is closely related to two other words that are related to δαιμόνιον: πνεῦμα, in that the "spirit" comes to rest on Jesus after he is baptized by John, and εὐθύς, because John speaks of making "straight" the paths of the Lord (e.g., Matthew 3:3 εὐθείας ποιεῖτε τὰς τρίβους αὐτοῦ). So at least with these two words, the semantic fields of δαιμόνιον and John the Baptist overlap with each other, which appears to be enough to make the rare word μαλακός appear in the most similar words for δαιμόνιον. The final word, ἕτερος, is used too diversely to easily recognize the reason it is considered similar to δαιμόνιον. More analysis would be required to determine this relationship.

If we look at these results in relation to the previous two tables, we should first notice that the average number of occurrences of the words in this list is 18.8, which falls about halfway between the average for Table 5.5 and Table 5.6. Though this number is still below the 25.55 average occurrences from the former table, it is close enough that it would require a broader analysis of other most-similar-word lists before coming to any conclusions about the types of words preferred by these two semantic extraction methods.

We should also note that the last two tables, which were the results of Word2Vec, have significantly more words that appear to be more tenuously related to δαιμόνιον than in the first table. If we look at the words in Table 5.5, we would consider 8 of the twenty words to have a real semantic relationship with

δαιμόνιον, either directly (βεελζεβούλ and ἄρχων), through being semantically related to the idea of exorcism (ἐκβάλλω and δαιμονίζομαι), or through being semantically related to the idea of miracle stories and sickness (κωφός, διαβλέπω, θεραπεύω, and νόσος). The other 11 words from that table for which we were able to find a distributional relationship to δαιμόνιον had this relationship because they just happened to occur within the context of exorcism or miracle stories or because, in the case of δοκός and σός, they co-occur with a word that is closely related to δαιμόνιον: ἐκβάλλω.

In the last two tables, however, we found only 4 words in Table 5.6 (λεπρός, δαίμων, χωλός, κακῶς) and 5 words in Table 5.7 (λεπρός, ἄλαλος, κωφός, χωλός, δαιμονίζομαι) that were semantically related directly to δαιμόνιον or to one of the semantically related spheres of exorcism and miracle stories and sickness. The other words seem to be related simply because they happened to appear in an important, semantically related context or to co-occur with a semantically related word. And though this is only a small sample of the data, this seems to suggest that Word2Vec, on a corpus as small as the New Testament, tends to be affected more by the relatively random occurrences of low-frequency words in important contexts than the Log-Likelihood method. And this observation perhaps goes hand-in-hand with the observation above that more lower-frequency words tend to appear on the Word2Vec lists than on the first list. If we were to continue our investigation of the results of these three language models, these would be thoughts that we should keep in mind as we move forward.

In the end, all the three language models returned the same central semantic representation of δαιμόνιον as a word that is related to Jesus' miracle stories and, thus, serves to demonstrate his power as a wonder worker. And even though we think that this central representation is most clearly shown in Table 5.5, the other two tables served to strengthen it by introducing important words that did not appear in Table 5.5, such as λεπρός, δαίμων, χωλός, κακῶς, ἄλαλος, and κωφός. We would also like to remind the reader here that the Log-Likelihood language model scored significantly worse on the Gap Score metric than either of the other models did. And despite this, it seems to have returned a clearer picture of the semantics of δαιμόνιον than either of the other two models.

4 Conclusion

In this paper we have demonstrated on the basis of a small sample of data the usefulness of having a more hands-on and task-related method to assess the

results of distributional semantic extraction algorithms. We discovered that for our relatively small corpus of the New Testament that the language model that scored the lowest on the Gap Score metric (the Log-Likelihood method) actually seemed to return the most straightforward representation of the semantics of δαιμόνιον. And even though it is possible that a broader investigation of the data would actually reveal that the opposite is true, we believe that we have shown that by taking the time to actually do in-depth analyses of the data returned by any algorithm, as we did in above, scholars will be better able to choose which algorithm and which parameters will return data that is most useful to their own purposes.

To actually put this assessment method into practice, we would suggest that a scholar choose a small and varied subset of words from their corpus that are as unrelated as possible to the subject under investigation to analyze. So if we were investigating the semantics of the word πίστις in the New Testament, the investigation that we carried out above could be useful since we would consider δαιμόνιον to have only a marginal semantic relationship to πίστις. If, however, we were investigating the concept of exorcism in the New Testament, then δαιμόνιον would be a poor word to choose since it has a very close semantic relationship with exorcism. The thought behind this restriction is that if one is trying to choose the best parameters for an algorithm by actually considering words related to the subject under investigation, one is likely to introduce one's own biases and expectations into the data production process. This is the reason behind the computational linguistic maxim of not training on the data that you wish to test.

We would also suggest choosing words from different syntactic categories (at least from noun, adjective, and verb) and with differing occurrence counts (some with high counts, some with low counts, and some in the middle). Such a wide variety of words will give a better picture of the algorithms and parameters under investigation than just looking at, e.g., frequently occurring verbs or infrequently occurring adjectives would. And we would also like to stress that this form of investigation does not in any way preclude standards-based testing, such as the TOEFL question test or the Gap Score test that we have used here. On the contrary, we believe that such testing is a precondition of being able to engage in the qualitative assessment that we propose here. One should first test the data using one or more such standardized tests and only then carry out a qualitative investigation of those models that look the most interesting. And finally we would like to stress that even though this qualitative assessment requires a significant amount of attention to detail, it is not as time- and labor-intensive as it might look. We were able to complete the qualitative part of the assessment of these three language models in approximately

16 hours, basically two full days of work. So if one did something similar for a list of 10 different words, one could expect to be finished with the qualitative part of the analysis in less than a month. And if this qualitative analysis comes at the beginning of a larger research project, then we believe that this time spent in quality control will pay dividends throughout the life of the project.

References

Bullinaria, John A., Levy, Joseph P., "Extracting Semantic Representations from Word Co-Occurrence Statistics: A Computational Study," 2007, <http://www.cs.bham.ac.uk/~jxb/PUBS/BRM.pdf>.

Bullinaria, John A., Levy, Joseph P., "Extracting Semantic Representations from Word Co-Occurrence Statistics: Stop-Lists, Stemming and SVD," 2012, <http://www.cs.bham.ac.uk/~jxb/PUBS/BRM2.pdf>.

Dunning, Ted, "Accurate Methods for the Statistics of Surprise and Coincidence," *Computational Linguistics 19*, 1993, 61-74.

Goldberg, Yoav, Levy, Omer, "Word2vec Explained: Deriving Mikolov et Al.'s Negative-Sampling Word-Embedding Method," *CoRR* abs/1402.3722, 2014, <http://arxiv.org/abs/1402.3722>.

Harris, Zellig, "Distributional Structure," *Word* 10, no. 23, 1954, 156.

Harris, Zellig, "How Words Carry Meaning", Language and Information: The Bampton Lectures, Columbia University, 1986, <http://www.ircs.upenn.edu/zellig/3_2.mp3>.

Jurafsky, Daniel, Marin, James H., *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Second Edition, Prentice Hall Series in Artificial Intelligence, Upper Saddle River, NJ: Pearson Education, 2009.

Louw, Johannes P., Nida, Eugene A., *Greek-English Lexicon of the New Testament: Based on Semantic Domains*, Second Edition, 2 vols., New York: United Bible Societies, 1989.

Manning, Chris, Schütze, Hinrich, *Foundations of Statistical Natural Language Processing*, Cambridge, MA: MIT Press, 1999.

Mikolov, Tomas et al., "Distributed Representations of Words and Phrases and Their Compositionality," *CoRR* abs/1310.4546, 2013, <http://arxiv.org/abs/1310.4546>.

Mikolov, Tomas et al., "Efficient Estimation of Word Representations in Vector Space," *CoRR* abs/1301.3781, 2013, <http://arxiv.org/abs/1301.3781>.

Mikolov, Tomas, Yih, Wen-tau, Zweig, Geoffrey, "Linguistic Regularities in Continuous Space Word Representations," in: *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings,*

June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA, 2013, 746-751, <http://aclweb.org/anthology/N/N13/N13-1090.pdf>.

Munson, Matthew, *Biblical Semantics Applying Digital Methods for Semantic Information Extraction to Current Problems in New Testament Studies*, Theologische Studien, Aachen: Shaker Verlag, 2017.

Nida, Eugene A., *Componential Analysis of Meaning*, The Hague: Mouton, 1975.

Nida, Eugene A., Louw, Johannes P., Smith, Rondal B.,"Semantic Domains and Componential Analysis of Meaning," in: *Current Issues in Linguistic Theory*, ed. Roger William Cole, Bloomington: Indiana University Press, 1977, 139-167.

Schnabel, Tobias, et al., "Evaluation Methods for Unsupervised Word Embeddings," in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, September 17-21, 2015, Lisbon, Portugal*, 2015, 298-307, <http://www.aclweb.org/anthology/D15-1036>.

Test Sets

For reasons of space and readability, this table includes only the number and letter identifying the domains. In order to find a fuller description of these domains, as well as all the words that belong to the domain, please visit <http://www.laparola.net/greco/louwnida.php>.

Domain 1	Domain 2	Domain 1 words	Domain 2 word
67 E	23 A	μακρός, χρόνος, πάντοτε	Ψωμίζω
14 F	74	φωτίζω, λάμπω, φέγγος	δύναμις
33 Q	16	διδαχή, σωφρονίζω, παιδεία	ἀποτινάσσω
15 D	23 C	ἐκβαίνω, ἐκπορεύομαι, ἀπολύω	Τεκνογονία
12 A	10 B	ἐλωΐ, θεός, αββα	γονεύς
12 A	11 B	ἄθεος, παντοκράτωρ, θεός	ἀδελφότης
20 C	6 P	ὄλεθρος, πορθέω, συναπόλλυμαι	ἄγγος
33 Q	28 C	παιδεία, κατηχέω, παιδεύω	φανέρωσις
4 A	33 O	ἀγέλη, ζῷον, θηρίον	ἀγγελία
37 A	13 D	συλαγωγέω, ἐνέχω, βραβεύω	Προγίνομαι
3 C	2 F	βάτος, χόρτος, σίναπι	Χαλκηδών
7 B	15 D	σκηνοποιός, σκήνωμα, σκηνή	ἀπέρχομαι
20 C	53 I	ὄλεθρος, συναπόλλυμαι, φθορά	προφῆτις
25 U	54	προμεριμνάω, μέριμνα, καταπονέω	ἀνάγω
43	87 C	ἀροτριάω, ἀμάω, σπείρω	Πρωτεύω
4 A	37 D	θηρίον, τετράπους, θρέμμα	ἡγεμονεύω

Domain 1	Domain 2	Domain 1 words	Domain 2 word
41 A	25 U	κατάστημα, ἀγωγή, διάγω	ἀμέριμνος
64	15 D	ἔοικα, ὁμοίωσις, ὁμοιόω	ἐκβαίνω
25 U	87 C	ἀμέριμνος, μετεωρίζομαι, προμεριμνάω	εὐγενής
23 A	37 A	βρώσιμος, τρώγω, βιβρώσκω	αὐθεντέω
85 E	30 A	ἐγκατοικέω, κατοικητήριον, κατοικία	Διενθυμέομαι
43	25 U	σπείρω, ἐπισπείρω, ἀμάω	Μέριμνα
87 C	33 J	ὑπεροχή, μεγιστάν, εὐγενής	ἑρμηνεία
53 A	67 B	ἱεροπρεπής, θεοσέβεια, εὐσέβεια	ὁπότε
67 I	8 B	φθινοπωρινός, θέρος, διετία	τρίχινος
28 E	23 A	κρυφαῖος, βαθύς, κρύπτη	Θηλάζω
37 A	23 G	βραβεύω, δαμάζω, εὐπερίστατος	Συζωοποιέω
28 E	11 B	βαθύς, ἀπόκρυφος, κρυφαῖος	Χριστιανός
14 F	64	ἐπιφαίνω, φέγγος, ἐκλάμπω	οἷος
67 B	53 A	πρότερος, πρῶτον, προφθάνω	Δεισιδαίμων
60 B	25 C	ἐννέα, δύο, τέσσαρες	φίλανδρος
49	28 C	σπαργανόω, ἐγκομβόομαι, ἐνδύω	ἀποκάλυψις
41 A	25 C	βίος, χράομαι, ἀναστρέφω	φιλόθεος
23 G	67 I	συνεγείρω, ζάω, ζῳογονέω	θέρος
23 G	24 F	ἀνάστασις, ζάω, ζῳοποιέω	παθητός
3 C	24 F	χλωρός, βοτάνη, βάτος	Πάσχω
54	33 E	ἀποπλέω, εὐθυδρομέω, πλοῦς	ἀπογράφω
33 F	67 I	προσαγωγή, λαλέω, προσλαλέω	Παραχειμάζω
67 B	5 A	προφθάνω, πάλαι, καινός	Διατροφή
20 D	30 A	κατασφάζω, θανατόω, σφαγή	ἔννοια
33 J	7 C	ἐπίλυσις, μεθερμηνεύω, ἑρμηνεία	Γαζοφυλάκιον
85 E	64	περιοικέω, κατοίκησις, κατοικία	τοιοῦτος
33 O	23 I	ἐκδιηγέομαι, ἀναγγέλλω, ἀγγελία	Λέπρα
10 B	74	μάμμη, ἀπάτωρ, προπάτωρ	ἰσχύω
8 B	5 A	θρίξ, μέλος, κόμη	Χόρτασμα
11 B	41 A	ἄδικος, ψευδάδελφος, Χριστιανός	ἀναστροφή
11 B	3 C	ποίμνιον, ψευδάδελφος, νεόφυτος	ἄψινθος
3 C	67 F	χλωρός, χόρτος, ἀκάνθινος	ἕως
23 C	15 D	γέννημα, τεκνογονέω, ἔγκυος	Μεταίρω
85 E	11 C	παροικέω, ἐνοικέω, περιοικέω	ἔθνος
53 I	85 E	ψευδαπόστολος, ἀπόστολος, συμπρεσβύτερος	Παροικέω
88 X	67 E	ὀργή, θυμομαχέω, θυμός	ἀεί
2 F	67 F	χαλκηδών, σμαράγδινος, ἴασπις	ἀναβολή

Domain 1	Domain 2	Domain 1 words	Domain 2 word
33 Q	7 B	παιδεύω, διδασκαλία, θεοδίδακτος	Κατάλυμα
10 B	6 P	προπάτωρ, πατρικός, πατήρ	Θήκη
23 A	3 C	βρῶσις, τρώγω, ἐσθίω	ἀκάνθινος
33 Q	43	παιδεύω, διδάσκω, διδασκαλία	ἐγκεντρίζω
74	88 X	ἰσχύω, κατισχύω, ἐξισχύω	ὀργίζομαι
53 A	6 P	ἱεροπρεπής, δεισιδαιμονία, θρησκός	ἄντλημα
24 F	4 A	πόνος, συνωδίνω, πάθημα	κτῆνος
2 F	85 E	σμάραγδος, χαλκηδών, χρυσόπρασος	οἰκητήριον
11 C	33 J	ἔθνος, πολίτης, ἐντόπιος	διερμηνευτής
20 C	57 H	ἐξολεθρεύω, ἀπόλλυμι, πορθέω	Παραδίδωμι
53 I	28 C	ψευδαπόστολος, εὐαγγελιστής, ἀπόστολος	ἐπίσημος
33 Q	4 A	θεοδίδακτος, παιδεύω, διδασκαλία	ἀλώπηξ
30 A	43	ἀναλογίζομαι, λογίζομαι, ἔννοια	Σπείρω
60 B	41 A	δέκα, ἑπτά, τέσσαρες	Χράομαι
23 I	6 Q	λεπρός, κάμνω, μαλακία	ὀθόνιον
67 B	2 F	προφθάνω, ὁπότε, ἔκπαλαι	σάπφιρος
13 D	87 C	ἐπεισέρχομαι, ἐπιγίνομαι, συγκυρία	Κυρία
67 I	12 A	παραχειμάζω, διετής, χειμών	Παντοκράτωρ
7 C	53 A	οἴκημα, κοιτών, ἀνάγαιον	θεοσεβής
20 D	88	θανατόω, κατασφάζω, ἀναίρεσις	Πταίω
67 A	30 A	προθεσμία, εὐκαιρέω, εὔκαιρος	λογισμός
28 E	67 A	μυστήριον, βάθος, ἀπόκρυφος	Προθεσμία
41 A	60 B	ἀναστρέφω, στοιχέω, προσφέρω	τέσσαρες
13 D	15 D	πληροφορέω, ἐνίστημι, προγίνομαι	ἐξέρχομαι
16	23 C	ἀπομάσσομαι, τρόμος, ῥιπή	ὠδίν
64	25 U	ὁμοίωσις, οἷος, ἔοικα	ἀνασκευάζω
4 A	14 F	ὑποζύγιον, ἄρκος, θρέμμα	ἐπιφαίνω
23 C	12 A	γέννημα, γενετή, ἔγκυος	Μαράνα
20 D	10 B	σφάζω, ἀποκτείνω, διαχειρίζομαι	πατρῷος
14 F	3 C	ἐπιφαύσκω, λάμπω, φῶς	ἄψινθος
11 B	25 C	ἐθνικός, ἐθνικῶς, ψευδάδελφος	Φιλία
33 J	20 D	διερμηνεύω, ἑρμηνεία, ἐπιλύω	Θανατόω
23 C	85 E	ἀρτιγέννητος, γέννημα, τίκτω	Περιοικέω
49	33 Q	περίθεσις, ἐνδιδύσκω, ἐγκομβόομαι	Διδάσκω
33 Q	57 H	θεοδίδακτος, ὑποτίθημι, διδασκαλία	Δόμα
25 C	53 A	φιλόθεος, φιλάδελφος, φιλόστοργος	εὐσέβεια
37 A	2 F	ζωγρέω, συλαγωγέω, συνέχω	ἴασπις

Domain 1	Domain 2	Domain 1 words	Domain 2 word
11 C	54	ἐντόπιος, πολιτεία, πολίτης	Πλέω
88	33 F	ἁμάρτημα, ῥᾳδιουργία, προαμαρτάνω	Λέγω
6 P	37 A	νιπτήρ, ἄντλημα, θήκη	Ζωγρέω
8 B	54	κόμη, κρανίον, ἔριον	εὐθυδρομέω
33 O	23 C	ἄγγελος, σπεκουλάτωρ, ἀγγελία	ἀρτιγέννητος
60 B	16	τρεῖς, ἕξ, ἑπτά	Σείω
16	53 I	τρέμω, ταράσσω, σαλεύω	ἀποστολή
12 A	24 A	αββα, κύριος, θεός	Βλέπω
15 D	8 B	ἀπέρχομαι, ἔξειμι, ἀποβαίνω	τρίχινος
15 D	67 A	ἀποβαίνω, ἐκπορεύομαι, ἐξέρχομαι	εὐκαιρέω