Vector hermeneutics: On the interpretation of vector space models of text

James E. Dobson
Department of English and Creative Writing, Dartmouth College, USA

Abstract

Scholars working in computational literary studies are increasingly making use of text-derived vector space models, by which I mean numerical models of texts that represent the distribution or modeled relations among the vocabulary extracted from these texts. These models, as this essay will argue, call for distinct modes of humanistic interpretation and explication that are related to but distinct from those that may have been used on the original source texts. While vector space models are analyzed using increasingly complicated quantitative methods and the explanation of their operation requires statistical sophistication, my emphasis on humanistic interpretation is quite intentional. This essay theorizes two major categories of vector space models, the document-term matrix and neural language models, to position these models as not merely descriptions of texts but inscriptive representational objects that perform interpretive work of their own in order to demonstrate the need for a multi-level hermeneutics in computational literary studies.

Correspondence:
James E. Dobson, Dartmouth College, Hanover, New Hampshire, USA.
E-mail:
james.e.dobson@dartmouth.edu

In this essay, I seek to bring ongoing discussions of hermeneutics in the humanities and modeling in science together through a theoretical account of model construction and interpretation in literary studies. I use literary studies, or rather computational literary studies (CLS), as the site for interrogating the meaning of modeling as this field sits at the juncture of these two disciplinary discussions (Piper, 2017; So, 2017).[1] Models are abstract objects. In computational fields, these digital objects are numerical representations, typically taking the form of a subset of encoded information, of some phenomena of interest. In CLS, such models are used as evidence for arguments about texts. But, in this humanistic field, they cannot function as evidence alone, for as this essay will argue, they are also instruments that shape both their interpretative possibilities and those of the textual objects that they represent. Vectorized representations of text are a special kind of digital object that model some of the possible meaning of textual sources while seeking to reduce the loss of information extracted from these texts. The significance of these models and the analysis of their operation serves here as a case study for a computational literary hermeneutics.

Scholars working in CLS are increasingly relying on text-derived vector space models. These models are produced from texts and represent the distribution of vocabulary extracted from these texts or modeled relations among the vocabulary found in the source texts. CLS scholars including Ted Underwood and Andrew Piper have used these large matrices of data to make arguments about the shape and significance of the conversion narrative and the persistence of key textual features correlated with genre. Cultural historians making use of CLS methods such as Peter de Bolla have analyzed change over time and semantic drift in

Digital Scholarship in the Humanities, Vol. 37. No. 1, 2022.
doi:10.1093/llc/fqab079 Advance Access published on 17 September 2021

the formation of complex concepts using vector models of historical data. These models have become the basis for higher-level analytical tools that are used to support claims about change over time, classify texts into genres, categorize possible topics within text collections, and provide proxy models of narrative plots, among several other contemporary uses (Jockers, 2013; Piper, 2018; De Bolla, 2019; van Eijnatten and Ros, 2019; Underwood, 2019). These types of vector space models are nonisomorphic inscriptive objects that call for modes of humanistic interpretation and explication that are related to but distinct from those that may have been used on their original source texts. While these models are analyzed using increasingly complicated quantitative methods and the explanation of their operation requires statistical sophistication, my emphasis on humanistic interpretation is quite intentional. Humanistic methods, including but not limited to hermeneutics, are needed to interpret vector space models of text sources. All vector space models, however, are not the same. This essay will first examine the document-term matrix and argue that because these vectors represent texts, they need to be interpreted in terms of the texts from which they are derived. I will then turn to a second major category, the word embedding model generated by the word2vec and fasttext package. These embedding models have a looser relation to their source texts and provide a different set of hermeneutical possibilities that are best described as grammatical in nature. Both of these model types, however, are not merely descriptions of texts but instruments that perform interpretive work of their own and a theoretically informed multi-level hermeneutics is necessary to make use of these models for CLS.

To begin to unpack, the consequences for the interpretation of vector space models, a short gloss of some key terms is necessary. *Vectors* should be understood as stored computable values that enable comparisons across a high-dimensional space of some number of texts. *Vectorization* refers to the process by which objects become represented as vectors; vectors stand in for the source objects. In the case of texts, vectorization involves the quantification of linguistic units and the values may include word frequencies, probability values, or co-occurrence relations, among other possible options. Vectors are defined more by the operations performed on them, the vector-wise numerical computations and comparisons, than their particular data structures and formats. While a single indexed object comprised some indeterminate sequence of values might be considered as a vector, it is the computational manipulation of multiple objects grasped in their entirety that defines a vector. *Vector space* is the large dimensional geometric space, the boundaries of which are defined by the included vectors, in which these objects exist.

The vector space models of interest to us are numerical models that locate either texts or words in a high-dimensional representational space. In a broader sense, we might consider the document-term matrix, which is essentially a table of word frequencies aligned by a common vocabulary so that each vector represents the distribution of this vocabulary within an individual text, a vector space model but the models with the greatest concern to hermeneutical theory are known as word embeddings—despite the name other segments of text may be 'embedded' in the same way as individual words—and are most commonly produced by a class of methods known as neural language models. Both models are created through instrumental means and involved the measuring (counting, comparing, predicting) of textual objects. They are inscriptions because the vectors located within the model, which is to say each text or word or other textual unit, are now represented by a particular geometric shape that is the result of an algorithmic transformation. In the case of a vector representing an entire text, the linear string of words has been decomposed into frequency values (potentially raw frequency counts, weighted values, or normalized by the other texts in the) and indexed by a shared vocabulary.

Vector models are produced through one-way transformations of text into another space. This was called the 'document space' in an early paper introducing the model and a document was defined by the following for each D_i document in a t-dimensional vector in which t represents the included vocabulary: $D_i = (d_{i1}, d_{i2}, \ldots, d_{it})$ (Salton et al., 1975). Vector space models are representations of text in that they provide an alternate mode of access to *some aspects* of the information contained within text. I use the term representation to gesture toward two aspects of digital or computational representation. The first is the more familiar sense with which we might describe any

digitized object as providing a representation of another object, through sampling and quantizing. Sterne (2012) suggests a more specific term, mediality, to describe digital objects, specifically in the relation between an audio recording and an MP3. What differentiates vector space data from the MP3 is the degree of information loss involved in text-based vectorization. While we might quibble about sampling rates and information loss in various audio encoding formats, the song remains the same. What characterizes mediation is the ability of one media, in the terms used by Bolter and Grusin (1999), to remediate another; we can produce a CD from a record and then convert the digital audio stream on the CD to another format. Vector-based representations of text are models of *some* of the information but not enough to reconstruct the text or create another digital object that might be said to be mediatic relation to the text. The other meaning of representation is more technical. This type of representation is that used in describing mathematical transformations. Take, for instance, the Fourier transform, an important operation for signal processing that decomposes numerical values to sine and cosine components. The transformed space provides a representation of the original values in a form that might be computational easier to compute than the original form. While the Fourier transform can be reversed and vector space transformations of text cannot, I want to capture this representational aspect in my reading of vector space models. While some theorists, most notably Johanna Drucker, want to displace the primacy of representation in reference to computational transformations in order to foreground the performative dimensions of knowledge production in these methods, in the case of text-derived numerical models it is the relation between the text and the model that authorizes any knowledge created by the model. This is not to say that the model is the same as the text, as I have mentioned above, these are nonisomorphic objects, but the meaning of the model rhetorically resides within modeled texts, especially with models of texts *qua* text rather than more general language models.

Recent debates about the capabilities required to interpret work in CLS arise from the fact that while vector space models are derived from text, textual analysis runs into limits when applied to numerical models. In an essay titled 'Is There a Text in my Data (Part 1): On Counting Words', Gavin (2020) makes the perhaps not surprising argument that word vectors are decidedly not texts. The stakes of this particular argument concern the validity of critiques from those not interested in mounting quantitative arguments or trained to create and analyze numerical data. These stakes are made explicit in the framing apparatus of Gavin's essay: he writes in response to Da's (2019) critique of CLS that appeared in *Critical Inquiry*. Gavin positions Da and Stanley Fish, who appeared as a respondent to Da's essay in a *Critical Inquiry* forum on the essay and who also provides the source reference for Gavin's title, as literary scholars trained to understand texts not data. Training and field specialization becomes key to Gavin's argument as he seeks to show that Da and Fish are not able to use their literary expertise to render a professional judgment of data objects. Gavin's defensive response to Da and Fish requires him to assert excessive limitations for the sort of interpretation practiced by literary critics when applied to data derived from texts. It is excessive because the majority of models invoked by both Gavin and Da cannot be understood outside of their relation to their textual sources. In separating the text from data, CLS risks dismissing a series of critical questions about these texts and their representation in numerical models.

The role of hermeneutics and the status of writing in the sciences is crucial to understanding how vector models fit into humanistic hermeneutical approaches. Two modes are helpful in delineating the status of writing within the sciences. On the one hand, we might understand writing in a literal vein and say that the sciences have only a tangential relation to writing in the form of grants, scientific communication in the form of peer-review publication, and debates involving the interpretation of experimental research. On other hand, we can consider the activity of science as aided by instruments as a mode of writing. This is the case for Bruno Latour who sees the laboratory as site of instrument-aided research that takes the form of inscription. Latour's account of instruments as writing is typically linked to the visual display of data, but we can take this more broadly to include the recording into data structures the instrumental aspects associated with vectorization including

measurement, feature selection, and normalization. Don Ihde has argued for an 'expanded' hermeneutics that can encompass the use of scientific observation and visualization. Ihde's (1998) account of a scientific hermeneutics is what he calls 'perceptually oriented'. It is focused on the reading of instruments, the viewing of imaging technologies including MRI data, and the interpretation of a graph. When we consider *in silico* experimentation, we invoke another mode of inscription, the creation of datasets, programs, and scripts that perform the work of computational science. Vector space models of written texts are tied up in these multiple aspects of writing but what differentiates these models from other numerical models is their relation to their source objects. Scientific imaging models may be considered isomorphic, which is to say that the created model visually resembles the object modeled and may be interpreted as such. Other models are nonisomorphic and are interpreted with more difficulty but read and interpreted nonetheless. As representations of representations, the vector space models under consideration in this essay are nonisomorphic objects that require interpretation in terms of the text that is the site of their genesis.

In the act of creating these data objects, CLS scholars model texts and relations among a group of texts but more importantly they are inscribing representations of these texts within their modeling. While they are composed as a set—the status of the set or collection as an important aspect of vector space objects will be elaborated later—of numerical values, vectors generated from textual sources cannot be conceptualized as merely quantitative data. They are more than mere 'samples' of textual sources. They neither are simply numerical renderings of text nor are they to be understood simply as text (Loukissas, 2019).[2] While vector representations might be generated from the results of earlier stage preprocessing or filtering procedures, they fundamentally remain numerical representations of information extracted from textual sources. This doubled representational quality has important implications for the interpretation of the values held within vector space and the possibilities for interpretations of the models themselves. While some might cast computational work in the humanities as part of the descriptive turn, the modeling of texts cannot be said to be descriptive for these methods are producing new inscriptions that are fundamentally interpretive and require interpretation in terms of the texts that have been modeled (Marcus et al., 2016).

Computational approaches to the study of culture are well suited to examining linguistic and semantic patterns in a large number of texts. Given additional metadata or descriptive dimensions such as a temporal labeling or annotation of genre or nationality, these same methods can tell us about statistically significant group-level differences in these patterns. Johanna Drucker inverts the expected comparison between statistical modeling and hermeneutics when she writes, in relation but not limited to her understanding of the graphical representation of data, of 'interpretation as probabilistic'. In this context, probabilistic means 'nondeterministic and selective' and Drucker extends this concept to encompass reading as such (Drucker, 2020). 'A text provokes a reading', she writes, and this reading 'allows it to come into being. Every reading is specific to its reading moment, it is an event generated between a reader and a text in a set of conditions never to be repeated' (44). While Drucker understands interpretation as a contingent event and thus not the proper name for algorithmic transformations of text, I am interested in drawing attention to the degree that interpretive decisions have been dismissed or made opaque in some computational work. An important question to be answered is whether it is the *use* of algorithms that is an interpretive or hermeneutic maneuver or if the *algorithm itself* is hermeneutic, which is to say that the algorithm performs some interpretative act on data. In short, both are hermeneutical operations. Interpretations derived from algorithmic manipulations of text are hermeneutical as are the underlying algorithms. In calling their computational tools 'hermenutica', Rockwell and Sinclair (2016) foreground this dimension of all computational transformations. Where we get into trouble is when we do not properly recognize that the algorithms that we use and models that we create are performing interpretive tasks. When CLS scholars create models, they are engaging in the interpretive inscription of a new object but they are also re-presenting these texts.

In order to think through the interpretation of vector space data in the humanities, we need to understand the relation between the texts to be modeled and the model. A relationship exists between the text and the vector space representation of the text but this

relation may not be exactly what we imagine it to be and this relation will depend upon the model type and parameters used. One way of thinking about vectorized renderings of text would be to posit, following the general logic of the digital, that these rows of quantitative values are 'samples' of the content of a text. This would involve something like the operation taking place when we apply a digital sampling algorithm to an analog audio recording or acquire a scanned image of print. These samples are judged as providing a good-enough representation of the content when they approximate the visual or aural experience of the 'original'. Yet, because vector space representations of texts are not explicitly designed for human consumption but for algorithmic manipulation, we cannot regard these values in similar terms (Burrell, 2016).[3] A vector space model cannot provide a variable rendering, which is to say a lower or higher resolution representation, of a text because these models are not primarily encoding a stream of sense data (as is the case with a MP3 file or a JPEG scan). These models are operating on and transforming already encoded data. The models, in typical use, are produced in exactly the same way with a scanned image of a printed text or a so-called 'born digital' text.

'What is at stake in vectorizing data?' asks Adrian Mackenzie. 'It produces a common space', he answers, 'that juxtaposes and mixes complex localized realities' (Mackenzie, 2017). It is this mode of inscription that happens simultaneously at the level of the text, the 'localized' reality in Mackenzie's formulation, and the common space—the common vector space resulting from decisions made about how to treat the localized individual text and parameters applied to one's preconceived notion of the common space. That aspect of vectorization, that the common space exists prior to the creation of the space as an imaginary construction is incredibly important to understanding and interpreting the results of vectorization. One way in which we see this common space defined is through parameterization, the selection of key thresholds and values. Parameters selected in advance of the transformation will determine how the individual vectors come together to form the vector space.

In one understanding of vector space, the vectorization of individual texts and the creation of the vectorized common space are separable. This is perhaps best illustrated through the example of the HathiTrust Extracted Features dataset. This dataset provides a standardized set of features from works in the HathiTrust archive, importantly including those works that remain under copyright protection. The extracted features from these copyright protected texts are able to be shared with scholars because they are not the text. They are designed for the so-called 'nonconsumptive' reading of machines rather than people. Because the extracted features cannot be reassembled to produce the original text, they are treated as data derived from text rather than texts themselves. The vectorization that results in the creation of the HathiTrust Extracted Features can be performed piecemeal rather than simultaneously across all texts, in fact with large collections it would not be feasible to do otherwise. When researchers build collections from these extracted features, they combine already vectorized data to create a new common vector space that is shaped by their selection criteria and preprocessing choices. Yet, this is not the defining moment in which a common vector space has been instantiated. That has already happened in a prior moment, in the selection of standardized methods used to produce the HathiTrust Extracted Features dataset. This selection involved an imaginary act of creation, imaginary because the common space was imagined as such even if no texts were vectorized and no code was executed. This is to say that vectorization is not a presentation of a collection of singular texts as data but an already complex representation of a common space. One may then ask, why was this particular representation chosen? Why not choose another representation? How was this choice made?

The document-term matrix or dtm might be said to compress, by way of the vectorization process described above, the possible semantic meaning of the text to the distribution of selected features and inscribe these into vector space. Vector models are also compressed because they lack a sense of time. Narrative time, as Jameson (1961) reminds us, is registered in a text's form, especially in the construction of sentences—are they long or short?—and punctuation marks that have the capability of quickening or slowing down our reading and thought (Jameson, 1961; Algee-Hewitt et al., 2017; Allison et al., 2017).[4] While vectorization has removed many of the formal features that we consider textual (word order, sentence, and paragraph structure), the document-term

matrix does retain the text's core vocabulary, which are interpreted back through the text. These matrices, I have been arguing, provide an alternative representation of *some* of the information contained within a text but as the primary meaning making unit of written language is usually taken to be the sentence and because document-term matrices dispense with word order and sentence structure, the 'information' recorded by any particular matrix is necessarily limited. While this information may be useful for comparing documents and determining the distribution of some features (i.e. individual words and more complex concepts and referents built from repetition of these words), the representations found in vectorized documents no longer have the ability to serve as sources for *some* arguments about the text documents and cannot provide adequate representations for many of important textual features.

Arguments about a single text, a text taken in isolation and removed from others in a set or collection, need to address the existence and composition of the collection itself, even if that collection is imaginary. This is to say that there is a crucial difference between making arguments about vectorized data from a single text and arguments that would insist on contextualizing a text with others from the same period, genre, author, etc. In the latter case, additional evidence is brought forward from contextual sources while in the former the availability of evidence is limited by the existence of the sources that define the collection. An example of this problematic can be found in computational stylistics. In stylistics, especially those studies in which the critic is interested in examining authorship, the data derived from text sources can only tell us about the likelihood of one person, identified as the author of one text, being the author of another text if both texts have been rendered as data into a shared space. The world of possible authors is determined by precisely those data derived from texts found within this shared space. There are no other potential authors conceivable within this framework.

The dtm, like all vector space models, is only made possible and sensible through the entire vector space itself. When a single vector is grasped, what is being grasped is more complex than an autonomous part taken from the whole, for the part in this case has meaning only in relation to the whole. This is true even if the values contained within the vector are simple word frequencies. The meaning of the individual vector is even more implicated in the geometry of the entire vector space if the vector values are scaled, normalized, or otherwise transformed into higher-level representations of the relationship between the term and the distribution of this term among all the vectorized documents that comprise this modeled space. It is because of this attribute of vector models that we should think of the dtm as a special kind of reinscription of the source text(s). It takes the form of a new object by providing a numerical representation of a text and includes as its own condition of possibility the presence of a collection of texts, even if that collection does not yet exist.

In their influential review article of vector space models, Turney and Pantel (2010) describe the representational aspect of these models: '[vector space models] extract knowledge automatically from a given corpus, thus they require much less labour than other approaches to semantics, such as hand-coded knowledge bases and ontologies' They continue, 'The vector $x:_j$ may seem to be a rather crude representation of the document d_j. It tells us how frequently the words appear in the document, but the sequential order of the words is lost. The vector does not attempt to capture the structure in the phrases, sentences, paragraphs, and chapters of the document. However, in spite of this crudeness, search engines work surprisingly well; vectors seem to capture an important aspect of semantics' (147). The crucial question, for humanists, is what exactly are these important aspects of semantics and how are these useful for understanding the selected texts? The answer depends, of course, on the construction of the entire pipeline: the algorithms, their parameters, the preprocessing methods used, any post-processing normalization, and even the documents supplied as input for the vectorization. Altering any of these components, any unit within the pipeline, can change the possibilities of the vector space and the semantic meaning of the vectors. Vectors may provide access to information derived from texts but what they primarily encode is the interpretive apparatus used to derive the vectors and this apparatus renders numerical information extracted from textual sources.

It is thus my argument that the creation of vector space models should be understood as an interpretation, which is to say a hermeneutical operation. When

we interpret and reinscribe a text as vectorized data, we are of course not reading in the sense that humans read. One should not confuse the lower-level computational functions, which 'read' the contents of stored data objects with reading. At the same time, human readers may not be able to interpret vectors in the same way in which we read a text and in our reading of computational forms of analysis, we do need to consider the specific affordances of these modes. Algorithmic manipulations of text, in short, are highly interpretive and while these procedures share some aspects of human reading, they are not the same as reading. We can better understand the stakes of such distinctions by examining the relationship between hermeneutics and reading. In an essay titled 'Why Distant Reading Isn't', Johanna Drucker makes the claim that reading is hermeneutic and what we call computational or machine reading should not be confused with reading because it is not hermeneutical:

> The distinction between mechanical and hermeneutic reading, between machine processing and cognitive engagement, between the automatic and the interpretative, between unmotivated and motivated encounters with texts, is essential. Processing is not reading. It is literal, automatic, and repetitive. Reading is ideational, hermeneutic, generative, and productive. Processing strives for accuracy, reading for leniency or transformation. No text-analysis program weeps when it reads the passages in Felix Salten's Bambi in which Bambi's mother dies (Drucker, 2017).

While this account leverages an effective response to reading for the force of its argumentation, Drucker's understanding of hermeneutics does not necessarily posit this possibility, readerly affect, as a requirement for reading. It is rather the contingent quality of engaged reading, what she elsewhere calls its probabilistic nature that defines reading for Drucker. Drucker's alignment of machinic with automatic produces a division that leads her to make the claim that computational text analysis does not perform interpretation. While we might want to put some pressure on her account of motivated versus unmotivated reading—it is certainly the case that there are motivations for the creation of the tools and in the particular instances in which they transform textual data. Therefore, machine reading, localized in this essay in the creation and use of vector space models, by necessity provides an interpretation of a text through its reinscription of this text into vector space and should also be recognized as a hermeneutical operation. Vector models are hermeneutical because, like all computational renderings of text, they are interpretive and because hermeneutics as such is not reducible to a single understanding; there are multiple levels hermeneutics and while we are not yet certain about the capabilities of some vector models like word2vec, they are certainly capable of performing certain lower-level interpretive moves through the modeling of discourse at the linguistic level from supplied language samples.

A Computational Theory of Hermeneutics

I want to now turn to a discussion of another category of vector space data, the kind produced by what are known as neural language models or embedding models in order to argue that while these models are highly interpretable by humanists they require multiple and potentially distinct horizons or levels of analysis. In order to advance this argument, I will examine the word2vec class of neural language models, a class that includes the quite similar Doc2Vec and FastText models. These models are popular, trivial to use, and frequently produce what appear to be highly interpretable results. The meaning of these results, however, is not as straightforward as they might initially appear. How might we apply hermeneutical modes of analysis to this type of object? In trying to understand the meaning of this type of vector model, we cannot simply follow the two major hermeneutical traditions: in one approach, the text is placed in relation to an author in the other, a context (Hirsch, 1967; Allington et al., 2016).[5] Neural language models may provide some context for understanding a text but frequently the statistical power requires comparing multiple larger models rather than the part, a singular text, in relation to the whole, or the entire model. What we need instead is a revised account of hermeneutics that takes into account the use and functioning of computational models

in order to understand the meaning of insights about language and texts that are derived from these models.

Motivating much of the use of more mature natural language processing techniques like collocation analysis and still emergent neural network-based methods of producing word embedding vectors is the widely shared contextual theory of meaning. The contextual theory of meaning, in short, posits that words found or 'embedded' in similar linguistic contexts, contexts here taken to mean relatively small windows of words positioned on either the left or right of a word of interest, as other words will have similar meanings. When Wittgenstein (2009) claims that 'the meaning of a word is its use in the language' or John Firth, a linguist, argues that 'you shall know a word by the company it keeps' we should not turn to computational extractions of uses that preserve and promote the most common contexts but rather understand these claims as provocations about the contextual meaning of language that imagines almost unlimited possible contexts for a word, contexts that extend beyond the past and contemporary and far into the future (Firth, 1957).

In many common implementations and parametrizations of word2vec, each vector is a 200 dimensional sequence of values that functions as a trace through the model and provides a location for each modeled word within a geometric space. Each dimension of this trace is a point in vector space. I use the term trace because it highlights the attributes of direction and magnitude that characterize mathematical vectors and it has resonances with the reinforcement strategies employed by neural networks. The vectors are inscribed within a space that they define but are also predetermined. Like the document-term matrix, each term is located in a geometrical relationship to the modeled vocabulary. Unlike the document-term matrix, each point corresponds to values located within a predefined number of dimensions, rather than being indexed by a vocabulary term. These geometrical relationships are incredibly difficult to visualize because of their high degree of dimensionality and are frequently scaled down to two or three-dimensional models in order to measure and inspect relations among terms, as determined by distance in the modeled vector space, or to find clusters of terms. Measuring pairwise distance, the distance in vector space between two individual word vectors, even at two hundred dimensions, is a trivial task but we must choose the appropriate method by which we determine the distance, our distance metric. Do we use cosine similarity and determine the angle corresponding to our two selected vectors? Or do we use Euclidean distance and measure the shortest path through vector space? When comparing multiple vectors for visualization in two or three dimensions we might choose principal components analysis or t-distributed Stochastic Neighbor Embedding to reduce the dimensionality of our data. These methods all produce alternate representations of vector space that introduce distortions and give greater significance to some aspects of data. They do not alter nor do they rewrite the vector space model but they do create a new representation of this space. The initial model and its geometry remain intact, but the space has been selectively extracted, extended, or warped to form a new representational object.

There is a way in which neural language models, in constructing geometrical models of language, both expose something fundamental about the interconnectedness of texts and language itself and limit the reshaping of language that takes place within text. Dominick LaCapra's gloss of the concept of text thus seems fitting to both aspects of textuality, those made apparent by the folding of neural network vectors and those unsettled relations of representation found within texts: '"Text" derives from *texere*, to weave or compose, and in its expanded usage it designates a texture or network of relations interwoven with the problem of language. Its critical role is to problematize conventional distinctions and hierarchies, such as that which presents the text as a simple document or index or a more basic, if not absolute, ground, reality, or context' (LaCapra, 1993). Neural language models weave a highly connected relational network in which all words or text fragments are placed in relation to each other. It is an instrument through which we can render language comparable but this comparison always takes place through the frame inscribed by the operator.

Vector space models, and in particular, those generated by the Skip-gram implementation provided by word2vec and its associated neural network algorithms, are representations of the distribution of words within training data. This distribution will

contain the particular way the selected collection of texts is composed, meaning the particular uses of language within these samples, and generalized features of the modeled language as language with little way to determine which network of relations have been registered by this instrument. Neural language models have proven themselves to be in some degree capable of modeling semantic relationships between words. Mikolov et al. (2013) demonstrate the capabilities of their algorithm and models with queries carefully constructed to show these capabilities. Similarity in vector space, where similarity means closeness as measured with one of the above-mentioned distance metrics within the model-defined geometrical space, has been assumed to suggest something about the similarity of the meaning of the words, or at least their usage patterns in supplied training data. These vector relationships prompt interpretive inquires and we understand the models and extracted vectors as meaningful objects. The 'space' of the word2vec vector space, like all vector space models, pre-exists the creation of the model itself and is therefore partially inscribed prior to training. This is to say that the model is instantiated or created with parameters that specify what will become the length of the vectors, the 'size' or dimensionality of the word vectors. This parameter provides one key factor for the creation of the vector space. The other key factor is the vocabulary size. This can be determined by the documents provided to the algorithm or regulated through a selection of model training parameters. The resulting vector space takes the shape of length of the vocab $\times$ length of vectors.

While the words within the model's vocabulary might be taken as arbitrary signs, the vectors themselves are quite removed from the properties of signifying language. These word-indexed vectors record relations based on common usage patterns. Much more complicated than chains of signifiers within what we might think of as conceptual range, vectors are the record of relations found among the vocabulary words as embedded within sample sentences. Relationships between words in the word2vec model are learned from the location of these words within the sentences provided as training data. The sentence is the core unit of the word2vec model. The window parameter for the model is designed to capture sentence-level relations between words. Examining word usage context within a window of ten words, for example, provides some expectation that each window contains distinct language patterns. Too small of a window and relations found in more complex constructions will be lost. Too large and the units of meaning making will be blurred and noise disrupts signal as unrelated words appear in close proximity to each other.

Because these vector space models render language and texts computable, it is tempting to read these according to structuralist theory. One might understand individual vectors within the document-term matrix as an utterance according the *langue-parole* distinction of Ferdinand de Saussure, or those generalized language features as langue and parole describing collection-specific features. The geometrical relations produced by word2vec lend themselves to reading according to these familiar paradigms, even if these multi-layer networks and high-dimension vectors, as odd as it sounds, flatten distinctions between language and collection features within the modeled texts. Lindgren (2020) conceives of word2vec as providing a model of discourse analysis comparable with the post-structuralist model provided by Ernesto Laclau and Chantal Mouffe. Lindgren takes a key term and renders it a nodal point embedded within a field of related signifiers; the words closest in space to the nodal point term forming then what he calls, after Laclau and Mouffe, the articulation of a particular discursive space. Stripped of the co-presence of other signifiers, the words necessary to train the model and provide the distances that make similarity possible, Lindgren reads the trained word2vec space as capable of sustaining analysis of a specific discourse contained within the broader language.[6] Yet filtering vector space to sift the 'signal' of the discourse of interest from the background 'noise' of the modeled language alters the significance of the model and the meaning of the highlighted vectors. While synonyms are found in a similar space, the word2vec model often results in parts of speech appearing together. It also results in colors and numbers appearing together.

A better theorization of vector space, one more attuned to the task of understanding the relationship between the individual vector and its containing and shaping space, can be found not in semiotic, structural, or even poststructural theory but in the hermeneutic

tradition. This framework provides the resources for thinking through both the complexities of vector space construction and the use of vectors. Hermeneutical theory is also especially able to address the ambiguous meaning of the type of vectors found within the word2vec model, in which we find both the similarity of type, such as numerals or plurals located in a similar space in the model, and concept similarity or synonym groups, like the similarity in vector space of various colors, red, blue, and green. In order to make this argument, we must first recognize that word2vec and similar neural language models are unable to independently model the supplied texts and the larger language system in which these texts are embedded. The theory of interpretation best able to address this problem and others found in contemporary computational methods applied to textual sources originates in 19th century biblical hermeneutics. Friedrich Schleiermacher's hermeneutics offers an unexpected starting point to begin this work of theorization. Schleiermacher was an early-19th century German theologian and scholar of the New Testament but his hermeneutic theories are his lasting legacy.

In several different monographs, Schleiermacher articulates what he terms general hermeneutics, which is to say a theory of interpretation that would be applicable to many different types of texts, not just the major religious texts of his discipline. In his account of this general hermeneutics, Schleiermacher makes a distinction between technical and grammatical interpretation that will have major implications for understanding neural language models like word2vec. Schleiermacher was especially concerned with written texts rather than spoken language. Written texts, he argues, in particular call for the use of his interlinked pair of interpretative practices. These two modes of interpretation inform the work of each other and assist the interpreter by delimiting the local task. While Schleiermacher's hermeneutics initially appear to be underwritten by the authority and presence of an author, such a figure is not necessary to the task of interpretation. His hermeneutics are textual but not directed only toward the explication of texts. When addressing the implications of Schleiermacher's hermeneutics for the computational modeling of textual sources, we can easily bracket his reliance on the figure of the author by examining the possibility of specialized models within larger language models.

Schleiermacher's distinction between technical and grammatical interpretation is fundamental. He argues that these two forms of interpretation are equal but distinct and both are required for hermeneutics. For Schleiermacher (1998), technical—or as it sometimes appears in his explication of the theory—psychological interpretation concerns individual utterances of authors. Grammatical interpretation, however, is concerned with the meaning-making operations of language as such. Grammatical interpretation, he writes, 'is based on the characteristics of discourse which are common to a culture; technical interpretation is addressed to the singularity, indeed to the genius, of the writer's message'. This distinction provides us with a way of framing different modes of vector space, even ways of understanding both as active within a single model. Larger neural network models derived from large textual archives of the word2vec category disable direct access to individual utterances. They produce useful results only because of their scale. Patterns learned by the algorithm arise from the accumulation of repetitions of words within similar windows. The individual text of Schleiermacher's 'technical interpretation' disappears and what we get in return is the promise of access to grammatical interpretation—indeed at a scale perhaps not before possible. Grammatical interpretation is a fitting name for the sorts of linguistic questions and queries typically applied to such vector space models. The analysis of texts requires both modes of interpretation and includes as the condition of its possibility the existence of a broad sampling of language. 'Not all discourse is the object of the art of explication to the same extent', Schleiermacher writes. 'Some utterances', he continues 'have a value of zero, others have an absolute value; most discourse lies between these two points' (13). The understanding of a specific discourse of interest requires mapping the language in which this discourse is embedded, even if that language at times is everyday language, highly repetitive, or appears to be lacking in significance for the project at hand. We ask, in many applications of these models, about the relationships between grammatical units. Searching for nearest neighbors in vector space is a grammatical operation. When use these tools, we are searching for information about typical usage patterns of a discourse that is common to the textual sources on which the model is trained, a general grammatical model of language.

Schleiermacher's first canon—he organizes his criteria for hermeneutical theory into ordered lists he calls canons—of grammatical interpretation states that 'everything in a given utterance, which requires a more precise determination may only be determined from the language area which is common to the author and his original audience'. Schleiermacher's second canon states that 'the sense of every word in a given location must be determined according to its being-together with those that surround it' shares with the claim for use-defined meaning in Wittgenstein and Firth but comes closer to the operation of word2vec in that it recognizes the embedded nature of our words, both forward and backward. The concept of 'being-together' suggest much more of a relational meaning among language than Firth's notion of context as 'company' for the target word. Grammatical interpretation is precisely the sort of hermeneutics both enacted by neural language models and made possible for the human interpreter equipped with these models as an instrument for understanding text. The vast geometries of word vectors are not primarily interpretable in terms of the text *qua* text but now in terms of the historical horizon of language captured by large numbers of these texts. The horizon is historical because these texts taken as a whole define what Schleiermacher calls the common language area.

In naming the word2vec vector space model grammatical, we can connect the expanded scientific hermeneutics that enables us to understand instruments as creating inscriptions to the functional capabilities of the neural language models identified above. Both of these understandings fall under the rubric of grammatology, as defined by Derrida (2016) in his 1967 *Of Grammatology*. Grammatology turns to 'inscription in general' to reconceptualize a whole range of activities and techniques and especially in those fields that have undergone a transformation to render their objects of analysis as carriers of information: 'It is also in this sense that the contemporary biologist speaks of writing and *pro-gram* in relation to the most elementary processes of information within the living cell. And, finally, whether it has essential limits or not, the entire field covered by the cybernetic *program* will be the field of writing'. Literary studies are still in the early stages of this sort of grammatization and the advent of text-derived computational models require rethinking the units of analysis. In using and interpreting these models, CLS scholars are engaging with forms of inscription that transform some aspects of writing and language into information; other aspects of writing escape the present informational schemes and certainly some of unable to become information.

The information modeled by the methods mentioned in this essay has different horizons of interpretation depending upon the model type. In the case of a model in which each vector provides a representation of a text, the vectors are grounded in and evaluated by their source texts. The vector is a part of the whole of the vector space; the entire space is a part of the whole of the training text documents. Despite their malleability following vectorization, the vectors in such a model remain transformed surrogates for the texts. The other primary model invoked, the word2vec embedding model, is also trained on texts but its vector space registers multiple aspects of language as found in the modeled texts. We might best think of this class of vectors as inscribed traces through their textual sources. The shape of these traces record repeated patterns of language use. Each trace, then, itself is a record, a line of inquiry and connection running through, constructing, and calling into being a semantic universe. To properly attend to these multiple layers of meaning woven through the model and its text sources, CLS needs the resources of a computational hermeneutics.

References

Algee-Hewitt, M., Heuser, R., and Moretti, F. (2017). On paragraphs. In Moretti, F (ed.), *Canon/Archive: Studies in Quantitative Formalism from the Stanford Literary Lab*. New York: N+1 Foundation, pp. 65–94.

Allington, D., Brouillette, S., and Golumbia, D. (2016). Neoliberal tools (and archives): a political history of digital humanities. *Los Angeles Review of Books*. May 1, 2016. https://lareviewofbooks.org/article/neoliberal-tools-archives-political-history-digital-humanities/ (accessed 18 February 2021).

Allison, S., Gemma, S., Heuser, R., Moretti, F., Tevel, A., and Yamboliev, I. (2017). Style at the scale of the sentence. *Canon/Archive: Studies in Quantitative Formalism from the Stanford Literary Lab*, New York: N+1 Foundation, pp. 33–63.

Bolter, J. D. and Gruisin, R. (1999). *Remediation: Understanding New Media*. Cambridge: MIT Press, p. 20.

Burrell, J. (2016). How the machine 'thinks': understanding opacity in machine learning algorithms. *Big Data & Society*, **3**(1): 1–12.

Da, N.Z. (2019). The computational case against computational literary studies. *Critical Inquiry*, **45**(3): 601–39.

De Bolla, P., Jones, E., Nulty, P., Recchia, G., and Regan, J. (2019). Distributional concept analysis. *Contributions to the History of Concepts*, **14**(1): 66–92.

Derrida, J. (2016). *Of Grammatology. Spivak, G. C. (trans).* Baltimore: Johns Hopkins University.

Drucker, J. (2017). Why Distant Reading Isn't. *PMLA*, **132**(3): 628–35.

Drucker, J. (2020). *Visualization and Interpretation: Humanistic Approaches to Display*. Cambridge: MIT Press.

Firth, J. R. (1957). *Papers in Linguistics 1934–1951*. London: Oxford University Press.

Gavin, M. (2020). Is there a text in my data? (part 1): on counting words. *Journal of Cultural Analytics*, **5**(1).

Hirsch, E. D. (1967). *Validity in Interpretation*. New Haven: Yale University Press.

Ihde, D. (1998). *Expanding Hermeneutics: Visualism in Science*. Evanston: Northwestern University Press.

Jameson, F. (1961). *Sartre: The Origins of a Style*. New Haven: Yale University Press.

Jockers, M. L. (2013). *Macroanalysis: Digital Methods and Literary History*. Urbana: University of Illinois Press.

LaCapra, D. (1993). *Rethinking Intellectual History: Texts, Contexts, Language*. Ithaca: Cornell University Press.

Lindgren, S. (2020). *Data Theory*. Medford, MA: Polity.

Loukissas, Y. A. (2019). *All Data are Local: Thinking Critically in a Data-Driven Society*. Cambridge: MIT Press.

Mackenzie, A. (2017). *Machine Learners: Archaeology of a Data Practice*. Cambridge: MIT Press.

Marcus, S., Love, H. and Best, S. (2016). Building a better description. *Representations*, **135**(1), 1–21.

Mikolov, T., Sutskeve, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pp. 3111–19.

Piper, A. (2017). Think small: on literary modeling. *PMLA*, **132**(3), 651–8.

Piper, A. (2018). *Enumerations: Data and Literary Study*. Chicago: University of Chicago Press.

Ramsay, S. (2011). *Reading Machines: Toward an Algorithmic Criticism*. Urbana: University of Illinois Press.

Rockwell, G. and Sinclair, S. (2016). *Hermeneutica: Computer-Assisted Interpretation in the Humanities*. Cambridge: MIT Press.

Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, **18**(11): 613–20.

Schleiermacher, F. (1998). *Hermeneutics and criticism: and other writings*. Andrew, B. (ed., trans.) New York: Cambridge University Press.

So, R. J. (2017). All models are wrong. *PMLA*, **132**(3), 668–73.

Sterne, J. (2012). *MP3: The Meaning of a Format*. Durham, NC: Duke University Press, pp. 9–10.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: vector space models of semantics. *Journal of Artificial Intelligence Research*, **37**: 141–88.

Underwood, T. (2019). *Distant Horizons: Digital Evidence and Literary Change*. Chicago: University of Chicago Press.

Wittgenstein, L. (2009). *Philosophical investigations*. Hacker, P.M.S. and Schulte, J. (eds., trans.). New York: Wiley-Blackwell.

van Eijnatten, J and Ros, R. (2019). The Eurocentric fallacy: a digital-historical approach to the concepts of 'Modernity', 'Civilization' and 'Europe' (1840–1990). *International Journal for History, Culture and Modernity*, **7**(1), 686–736.

Notes

1 There has been limited work on modeling in CLS Two important essays on modeling appeared in the same issue of *PMLA*.

2 For an argument of visualizations as texts, see Loukissas (2019), in which he presents visual data and then argues that 'in engaging these visualizations, the reader should be ready (as they must with any evidence) to do some of their own interpretive work Visualizations are, after all, also texts' (8).

3 In addition to the data format and models, we should also consider the output of the algorithms themselves Jenna Burrell provides a compelling account of how some machine learning algorithms, especially when applied to some kinds of data, are not designed for interpretability by humans but rather for accuracy on a task. Burrell writes: 'When a computer learns and consequently builds its own representation of a classification decision, it does so without regard for human comprehension. Machine optimizations based on training data do not naturally accord with human semantic explanations'.

4 Jameson's (1961) account of narrative time in relation to sentence and paragraph structure can be found in *Sartre: The Origins of a Style* There have been some

computational and formalist accounts of style focused on sentences and paragraphs. See Mark Algee-Hewitt et al. (2017)

5 Some computational work within literary studies, especially those making using of author-focused stylistics, return to prior modes of scholarship in which the author of the texts becomes determinative One notable exception to the shift away from author-centric models in 20th-century hermeneutical theory is E.D. Hirsch's (1967). Allington et al. note several commonalities and connections between what they identify as the motivating ideology within the digital humanities work and Hirsch.

6 Lindgren provides several visualizations of what he terms the discursive space of key concepts in his analysis of Reddit discourse His comparison of the word2vec vector space and the positioning of vocabulary as nodal points within the space with Leclau and Mouffe's use of Jacques Lacan's *point de capiton* in their co-authored *Hegemony and Socialist Strategy* (1985) is provocative; yet, it is not clear that the concept of a master signifier can function within the logic of this unsupervised model. Lindgren also drops the explicitly post-structural framing within the theory that draws attention to the undecidability of the discursive field.