Kant be understood? Probing the parameters of semantic models of philosophy

Brent Kievit-Kylar[1] and Colin Allen[1]
[1] Indiana University Bloomington, Indiana, U.S.A
bkievitk@indiana.edu, colallen@indiana.edu

Abstract.

The discipline of philosophy is blessed with a number of relatively mature digital resources. Consequently, philosophers are well positioned to exploit algorithmic methods for discovering and analyzing meaningful relationships in and among large bodies of text. Successful exploitation will depend, however, on deepening our knowledge of what these algorithms do and how they manage to do it. To move from digital philosophy to computational philosophy, it is necessary to go beyond the production of intuitively plausible results, towards more systematic comparative investigation of the computational models applied to philosophical texts. Algorithmic learning models allow computational systems to automatically accumulate vast quantities of relational data that are proxies for the semantics of words. As with all models, parameters must be set and assumptions must be made. But how much do these choices affect the resulting output and therefore the conclusions of the modeler? Furthermore, for philosophy there is nothing resembling a gold standard to which results may be compared. In this paper, we explore how the "black art" of parameter setting in these semantic models affects the outcomes in a comparative study, using the Stanford Encyclopedia of Philosophy as the data source.

Introduction

Semantic space models, developed by computer scientists and computational linguists, represent words or documents as points or vectors in a multidimensional space. There are several techniques for training such models on text corpora. For different families of models, the dimensions of the space may correspond to individual lexical tokens or be arbitrarily assigned. Semantic space models support complex analysis by acting as a proxy for the semantics of words. Their use allows modelers to explore differences within and between corpora that may be too large for a single individual to fully grasp. They also provide a systematic analytical tool, unbiased by external human preferences.

The application of semantic models to philosophical sources has begun to produce intuitively plausible results, such as visualization of how different online encyclopedias locate prominent philosophers differently with respect to a network of concepts (Allen et al. in press). However, systematic investigation of how model parameters affect outcomes within a given domain is lacking, not just within philosophy. Modelers are faced with choices not only between different models, but about how to set up the models prior to training them on a particular corpus. Many decisions must be made, such as which words to place on a stoplist, how many dimensions to use for the vector space, and how many topics a topic model should extract. Currently, these decisions constitute a kind of

"black art" in which parameters are tweaked to produce plausible looking, even useful, results for release to somewhat appreciative audiences. In addition, there has to date been very little systematic exploration of the robustness of the results obtained by these methods.

We believe that deepened understanding of what the models are telling us about the underlying corpora is necessary if their results are to be used more effectively. In this paper, our goal is to explore the parameter space of one such model, the BEAGLE (Bounded Encoding of the Aggregate Language Environment) model of Jones & Mewhort (2007). BEAGLE is of particular interest because it has proven successful in capturing human judgments of term similarity (Jones & Mewhort, 2007). Its application to a more specialized domain has not, however, been investigated until now.

A complete investigation of these questions involves more than can be covered in a single paper. Here, as an essential step towards further investigation, we start with a narrower question: How do different model settings affect the representation of philosophers and their ideas in different instances of the BEAGLE model trained on the SEP corpus? We center our current investigation on the representation of 'Kant', a philosopher, whose name is mentioned over 4,000 times in the SEP in 39% of the entries (not counting bibliographic references), a frequency exceeded only by 'Aristotle'.

1 The Corpus

For all experimentation in this paper, we use the Stanford Encyclopedia of Philosophy (SEP), an online, open access reference work written and refereed on a volunteer basis by over 1,500 professional philosophers under the guidance of two paid editors. As of the end of 2012, the SEP contained almost 17 million words distributed across over 1,300 articles. SEP articles are deliberately designed to provide a synoptic overview of the current state of academic philosophy. However, like all such endeavors, the goal of strict neutrality or unbiased representation is more easily stated than achieved. Nevertheless, encyclopedias provide an interesting test bed for model comparisons precisely because of their comprehensive goals.

2 BEAGLE

BEAGLE is an automatic, language comprehension tool, designed on the principle that a word is known by the company that it keeps (Firth 1957). When given a corpus, BEAGLE first splits the data into paragraphs, sentences, and then words. A moving window defined by a fixed number of words, or grammatical divide (sentence or paragraph) is then used to update the representation of each word with information about its neighbors. Each word is represented by two vectors, an environmental vector and a lexical context vector. (For present purposes we ignore a third, "order" vector that is part of the full BEAGLE model.) The environmental vector is generated upon the first encounter with a new word. Its values are each chosen independently from a Gaussian distribution. The dimensionality of the vectors is a parameter chosen by the modeler. Once chosen, this vector will remain constant and associated with the same target word every time this word is re-encountered. The lexical vector contains the learned information about the corpus. It is updated for each word encountered within the sentence or fixed window, by adding the lexical vector of the co-occurring word, with the environmental vector for the target word. Similarities between words are then calculated as the cosine between the lexical vectors.

Another way of looking at the BEAGLE model is based on a co-occurrence matrix. If we have a co-occurrence matrix M where $M_{x,y}$ is the number of times that word X, has co-occurred with word Y within the given window, we can generate BEAGLE models from this matrix. This is because the end value of the BEAGLE lexical vector is order independent (because it is only a series of additions). To

generate a BEAGLE model from a co-occurrence matrix, we must first generate an environmental vector for each word in the model. The lexical vector for each word can then be set to the weighted sum of the environmental vectors for every other word (where the weight is equal to the co-occurrence of the two words).

3 Parameters

For all model runs, the corpus was trimmed by removing header and footer information and web specific tags from every entry. Low content words (such as "the", "a", "is", etc.) were removed using a stoplist (see http://inpho.cogs.indiana.edu/datablog/ for more details). Given a corpus trimmed in this way, two primary factors influence the behavior of the BEAGLE model. The first is how the data are cleaned. Cleaning involves normalizing the surface structure of the text, such as removing non alphanumeric characters and normalizing letter case, as well as applying stemming (which operates without context to reduce inflected forms to their stem or root form). The second is the parameter settings of the model, primarily dimensionality and window size.

3.1 Letters

All letters in the corpus were cleaned using a whitelist (all letters not in the set were replaced with a space) using two different whitelists. The strong approach left only the letters a through z, while the weak approach included the following letter-like characters "-'îìëêéçæåäãâáÿµšž".

3.2 Stemming

Another optional cleaning procedure tested was stemming. Our implementation used the snowball stemmer (http://snowball.tartarus.org/). This procedure removes stems and postfixes such as pluralizations which can obscure the underlying similarities between words

3.3 Window Size

The window parameter represents the size of a chunk of text that the model uses when adding environmental vectors of neighboring words into the lexical vector representing a word. Windows that are too small do not allow sufficient exposure to the statistics in the corpora and will not find long range dependent connections between words. Windows that are too large associate words that are not actually proximal and can cause spurious similarities. We explore one- and two- and four-word windows where each word learns about those words that are one and two and four words adjacent. We also explore windowing based on sentences, where sentence breaks are indicated by punctuation (periods, question marks and exclamation marks) and paragraphs.

3.4 Dimensionality

Because the environmental vectors are generated from a Gaussian distribution at the beginning of the model run, the classic BEAGLE model is non-deterministic (in that different runs will give different environmental vectors, resulting in different similarity measurements). BEAGLE vectors remain a constant size throughout the model's lifespan, but the exact value of this parameter is decided by the model creator. Larger sizes will clearly take more time and space to compute, but it has not been well studied how a change in size affects the output of the model.

Another important question about dimensionality is whether increasing the dimensionality toward infinity will cause the model to approach a single set of similarities or if there might be multiple attractor states in which different sets of similarity can be reached from different starting locations. It turns out that this question can be answered analytically, and there is a single set of similarities toward which all models approach. To find this solution, we calculate the expected similarity of the cosine distance between any two lexical vectors. Expected similarity roughly translates to the average similarity of all possible starting environmental vectors. On the basis that the expected product of two different random Gaussian variables is 0, and the square of one random Gaussian variable is 1, we can quickly solve for the expected similarity of the cosine between any two lexical vectors.

We begin by remembering that each lexical vector is, in the limit, equivalent to a weighted sum of environmental vectors. From this, it follows that the expected cosine similarity between any two BEAGLE lexical vectors is equal to the cosine between the co-occurrence vectors for those two words. For the remainder of this paper, we will refer to this expectation model as the gold standard. This is not an indication that these are in any way guaranteed to be the correct similarity measures between words (especially as they still depend on how the corpus is parsed, what the corpus contains, and the assumption that this is a correct measure of word similarity). Instead they are to be thought of as gold standards only within the domain of the BEAGLE models. They represent the model that other BEAGLE models (with the same parameters) are approaching.

4 Visualizations (Word 2 Word)

Our explorations in this paper used the Word 2 Word graph visualization tool (Kievit-Kylar & Jones 2012). In Word 2 Word, words can be represented by nodes and word similarities can be represented by edges. The resulting graph can then be visualized in different ways. Word 2 Word allows users to select among different types of models, set parameters, select a data source, and train the model. Given a trained model, it allows users to select a subset of words, arrange them in various ways on the screen, and apply various data statistics to the resulting graph.

5 Word Similarity Over Set Size

An important comparative question to ask about two models is how similarly they treat a given word. For example, given that 'Kant' is an important philosopher found in many contexts within the SEP, how similar are different representations of the semantic space of 'Kant' between different model instances trained on the SEP corpus?

To answer this question, we calculate the degree of similarity between representations of a target word between two models. This is a three-step process. First, we select the two within-model similarity metrics that we are interested in. In the present study, because we are comparing different runs of the BEAGLE model, we use cosine similarity within each of the trained models. Second, we select the set of words in the neighborhood a given target word that we are interested in comparing. There are many ways of selecting such a neighborhood, but we focus on four different algorithms. Each algorithm takes as input two sets of word/value pairs, representing the similarity, according to the chosen metrics, of the target word (such as 'Kant') to each other known word in the trained models. As output, it returns a list of words. This list is parameterized by N, in which N is the number of words returned. Each of the four algorithms is described below.

Union: Select the top N most similar words to the target word from the first model according to the chosen similarity metric, then select

	the N most similar words to the target word from the other model according the similarity metric chosen for it. The word set is the union of these two sets.
Intersection:	Collect the two sets in the same way as in the union, but then take their intersection as the new word set.
First:	Select the top N most similar words to the target word according to the similarity metric from the first model only.
Average:	Select the top N most similar words to the target word according to the average similarity according to the metrics for each model.

At the third step, we generate a list of triples for each word selected in step 2, such as (plato,.2,.1) which indicates that the similarity between 'Plato' and 'Kant' for the similarity metric applied to the first model, is .2, and for the similarity metric applied to the second metric, is .1. We then compute an overall list-level similarity value between the two lists. In this paper we investigated three methods for assessing list-level similarity between the models:

Spearman rank:	An ordinal measure of the similarity of two orderings. Spearman sums the relative difference in order values for each order pair.
Kendall Tau:	Another ordinal measure, this similarity computes the number of concordant pairings.
Pearson coefficient:	A non-ordinal metric, this similarity computes linear dependence between two sets of values.

Each of these similarity measures returns a value of similarity between the two lists given. An interesting question is how list-level similarity differs as a function of N used to select the list of neighbors. Thus, we plot N against the similarity ratings obtained.

6 Procedures

6.1 Between Words

To understand how different words are represented within separate instances of the BEAGLE model, one must first gain an understanding of the similarity metrics. In this section we gloss over the parameters within the BEAGLE model to focus on word-selection differences. To this end, all models were trained on the weak whitelist data set, without stemming, using the sentence as a window size and using the gold standard for dimensionality (i.e., effectively infinite). The word-similarity-over-set-size metric has four by three, or twelve different combinations of word selectors and similarity metrics, as per steps 2 and 3 in section 6.1. How different are these metrics and how much trust can be put in the similarity due to parameter setting versus similarities actually inherent in the data? To provide a sense of the answer to this question, Figure 1 shows the results of running all of these comparisons. In this particular example, we compare the context of 'Kant' to the context of 'Fichte' (appearing 272 times in the SEP corpus). The rows represent the selection method used (Union, First, Average, Intersection), and the columns represent the list-level similarity measure used (Spearman, Kendall, Pearson). Each graph shows the value of N for the number of terms compared on the X-axis (zero to 10,000) and the similarity values obtained on the Y-axis.

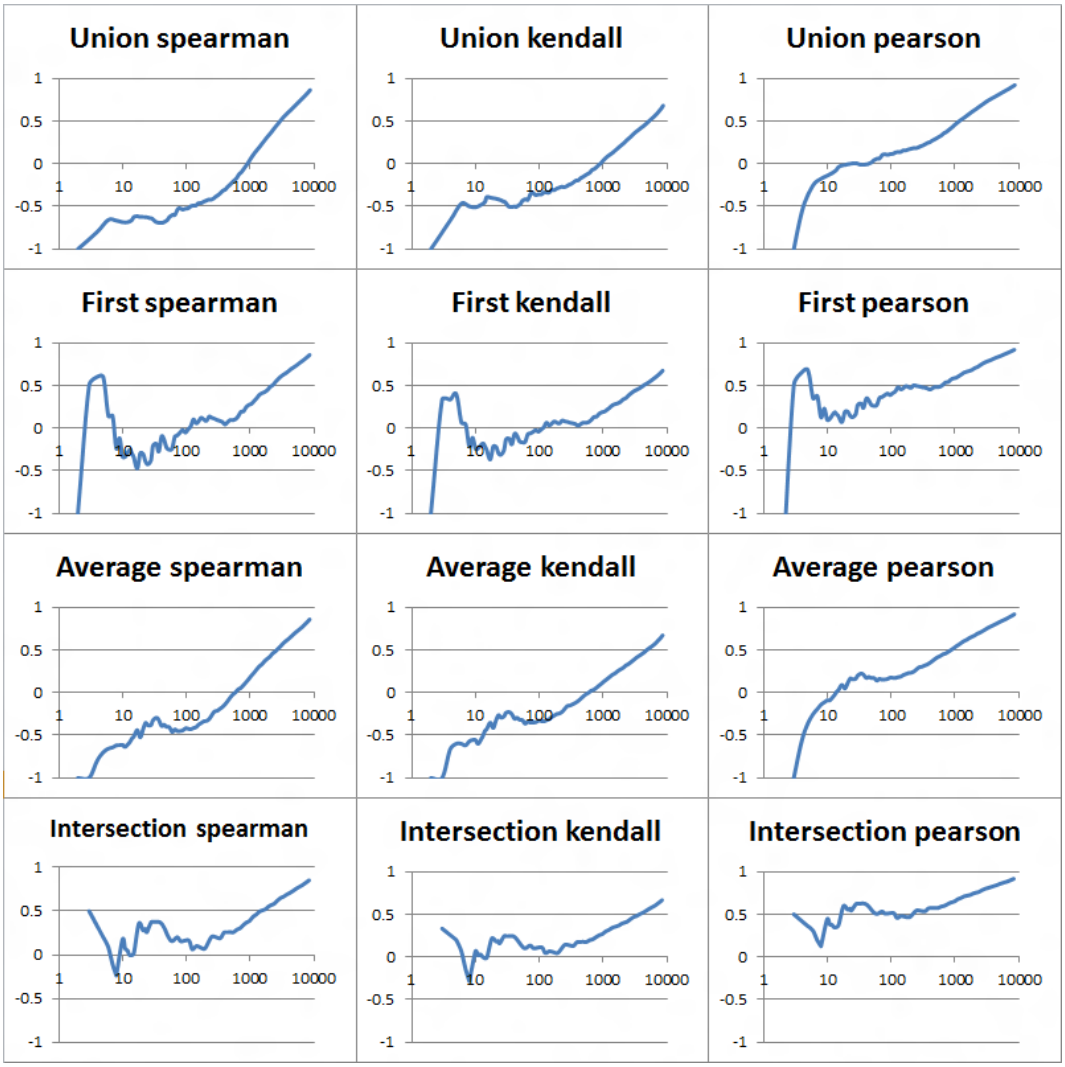

Figure 1. Constant: Similarity of the context of Kant to the context of Fichte using different neighborhood selection and similarity measures.

It is important to note the high level of uniformity in the shape of the graphs for each similarity measure. Each of the different neighborhood selection tools generates a differently shaped graph but the similarity metrics maintain shape to a great extent. It is also important to note that, while the similarity ratings vary greatly for values of N between 1 and around 300, from this point on, there is a relatively consistent linear slope visible. At the tail end of the distribution, there is nothing special about the choice of 'Fichte'; similar results hold for 'Goodman', e.g.

What more can be said about the different shapes within and between selection methods? As more and more terms are added into the analysis, it is not surprising that differences between the orderings produced by the models should tend to wash out. But cognitively, this is less interesting than the structure shown over the first few hundred terms. For readers of the SEP, judgments about how well a particular instance of BEAGLE captures the semantic space of a name such as 'Fichte' or 'Kant' is likely to be judged by the way in which the model relates these names to the few dozen nearest terms

in the semantic space, rather than thousands of terms. The fact that in three of the four metrics there is a high degree of list-level similarity for low values of N may reflect the similar way in which thinkers are initially introduced and discussed in the SEP.

Despite some differences at low N values, in the long run, the different similarity metrics tend to give similar results. Thus we use the union selection and Pearson correlation options for all the subsequent data comparisons.

6.2 Exploring the parameter space

Dimensionality of the BEAGLE model can have a significant effect on the semantic spaces produced by that model. Figures 2 and 3 show how dimensionality affects the similarity of the semantic space around the word 'Kant'.

Figure 2 represents the answer to the question, how different would the models have been different if different numbers of dimensions were used? Random variation in the environmental vectors was controlled for by training the lower and higher dimensional models using the same environmental vectors, but with dimensions removed for lower dimensional versions. In figure 2, the x and y axes represent different dimensionality values for the compared models, and the height or z axis represents the correlation between the similarity word list of 'Kant' in each of these models (using the between-words procedure discussed above).

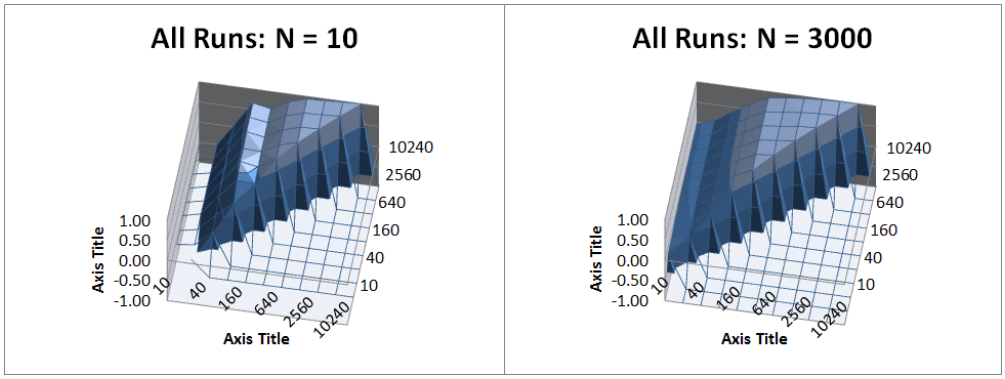

Figure 2. Comparing different dimensionalities. Surface plot (One run)

Figure 2 indicates that higher dimensional models show less between-model variation, but do these models deliver relevant results? To help answer this question, Figure 3 shows the 10 most similar words to 'Kant' for each different dimension setting. Color choices are arbitrary but indicate words which occur in multiple lists. The increasing overlap of most similar terms towards the right side, and the intuitive plausibility of the terms listed there, suggests that modelers seeking to understand the semantic space of philosophy using BEAGLE should set the dimensionality of the model to at least 1,000, although further investigation is warranted.

20	80	320	1280	5120	gold
developments[0.87]	spinoza[0.72]	offers[0.70]	hume[0.68]	hume[0.68]	hume[0.67]
implications[0.85]	show[0.71]	traditional[0.69]	traditional[0.65]	argues[0.65]	traditional[0.66]
foundations[0.85]	views[0.69]	hume[0.68]	found[0.63]	traditional[0.65]	argues[0.65]
close[0.84]	aristotle[0.67]	spinoza[0.66]	spinoza[0.62]	found[0.65]	locke[0.64]
postcolonial[0.83]	based[0.67]	presents[0.65]	argues[0.62]	locke[0.64]	found[0.64]
rooted[0.82]	analogy[0.67]	argues[0.65]	locke[0.62]	moral[0.64]	contrast[0.64]
humean[0.82]	favor[0.66]	views[0.62]	moral[0.61]	contrast[0.64]	moral[0.64]
husserl[0.81]	offers[0.66]	begin[0.62]	claims[0.61]	based[0.63]	based[0.63]
applications[0.79]	appeal[0.65]	locke[0.62]	based[0.61]	finally[0.63]	finally[0.63]
platonism[0.79]	intuition[0.65]	key[0.62]	finally[0.61]	regard[0.63]	descartes[0.63]

Figure 3. Comparing different dimensionalities. Top 10 most similar words for 'kant'. Color indicates multiple occurrences (One run)

Figure 4 shows a word centered layout around 'Kant' where the similarity measure for each of the dimensions are shown overlapping. Similarities that are greater than .7 are shown. Line color indicates the dimensionality of the model that produced that similarity with lighter lines belonging to models with higher dimensionality.

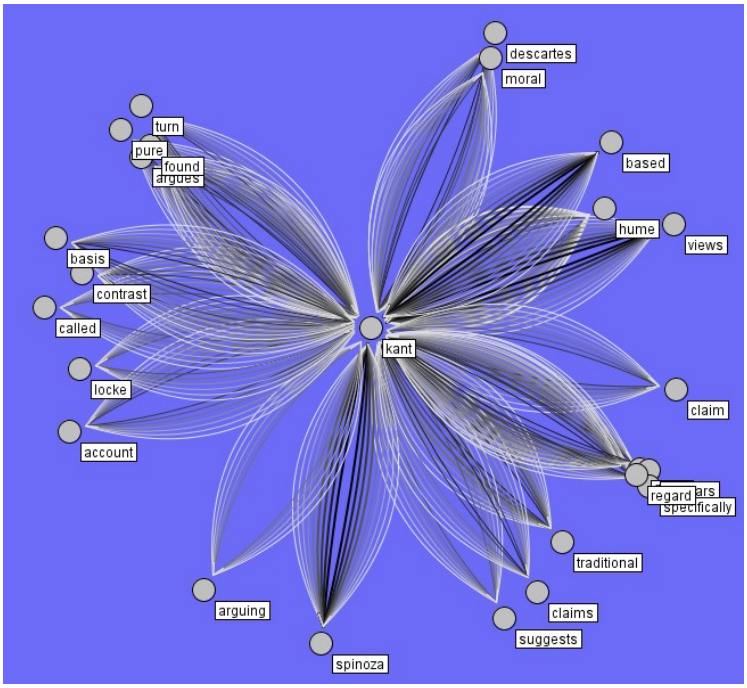

Figure 4. Showing the Kant similarity space for all different dimensionalities. Lighter lines indicate high similarity in higher dimensional space. (Distance to Kant indicates similarity of the word, relative angle is arbitrary)

Figures 2-4 indicate that models with dimensionality higher than a thousand will produce almost identical similarity measures. These measures also share an extremely high correlation with the gold standard. Therefore, for the rest of the paper, we will use the gold standard for other comparisons.

The next important hidden variable to semantic space models such as BEAGLE are the various word cleaning measures implemented prior to model training. We explored a 2 x 2 space of stemming compared to non-stemming, and a strong versus weak letter filter. Figure 5 shows the top 20 most similar words to 'Kant' under each of the four test conditions. The color coding indicates words which appear more than once on this list.

No Stems - Weak	Stemmer - Weak	No Stems - Strong	Stems - Strong
kant's[0.70]	critic[0.77]	hume[0.67]	critic[0.77]
argues[0.62]	hume[0.76]	traditional[0.66]	hume[0.76]
finally[0.60]	contrast[0.75]	argues[0.65]	contrast[0.75]
hume[0.60]	reject[0.75]	locke[0.64]	reject[0.75]
claim[0.60]	argu[0.75]	found[0.64]	argu[0.75]
contrast[0.60]	suggest[0.74]	contrast[0.64]	suggest[0.74]
regard[0.60]	articul[0.74]	moral[0.64]	articul[0.74]
reason[0.59]	lock[0.73]	based[0.63]	found[0.73]
descartes[0.59]	found[0.73]	finally[0.63]	lock[0.73]
specifically[0.59]	oppos[0.72]	descartes[0.63]	oppos[0.72]
claims[0.59]	claim[0.72]	regard[0.63]	claim[0.72]
fact[0.59]	moral[0.72]	appears[0.63]	moral[0.72]
found[0.59]	defend[0.72]	claim[0.62]	posit[0.72]
thinks[0.58]	ground[0.72]	suggests[0.62]	ground[0.72]
locke[0.58]	posit[0.71]	arguing[0.62]	metaphys[0.72]
moral[0.58]	philosoph[0.71]	specifically[0.62]	defend[0.71]
account[0.58]	metaphys[0.71]	account[0.62]	turn[0.71]
idea[0.58]	turn[0.71]	turn[0.61]	philosoph[0.71]
pure[0.58]	regard[0.71]	views[0.61]	regard[0.71]

Figure 5. The top similarities to the word Kant for each different cleaner. Each word that appears more than once is color coded with an arbitrary color to make comparison easier. Numbers in brackets show similarity.

One of the biggest difficulties in comparing results between these different conditions, is that not only are the similarity values between the same words different, but the possible words to compare to may be different. For example, in the no stems/weak condition, the word "Kant's" is the second highest match with 'Kant'. This word cannot appear on any of the other lists as both a stemmer and the stronger word filter will exclude this word. Nevertheless, visual inspection of these results suggests that stemming seems to have a greater effect on the word similarities than the letter filter. Figure 6 confirms this visual expectation by exploring the word similarity versus set size over all four of the pre-processing conditions.

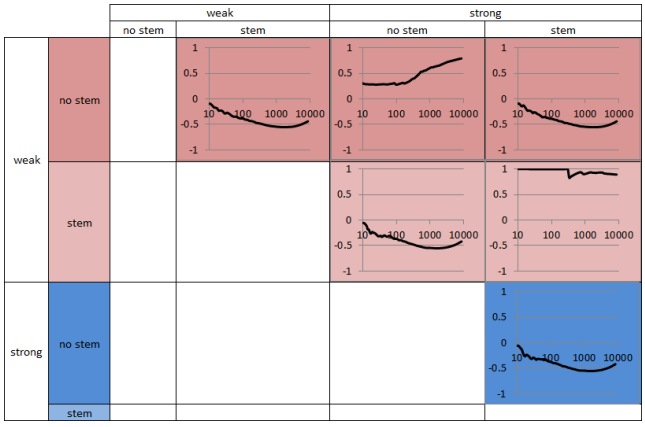

Figure 6. Similarity of the context of Kant with different cleaning procedures. The two cells with higher lines indicate higher similarity despite the different stemming procedures.

Figure 6 shows an almost perfect correlation between the stemmer-weak and stemmer-strong condition as well as a strong correlation between the no stemmer-weak and no stemmer-strong conditions. Other conditions have almost no correlation across the board.

The final variable tested was window size. This is the unit size of the number of words that were learned together. Figure 7 shows word similarity for 'Kant' across 5 different window sizes (1, 2, 4, sentence, paragraph).

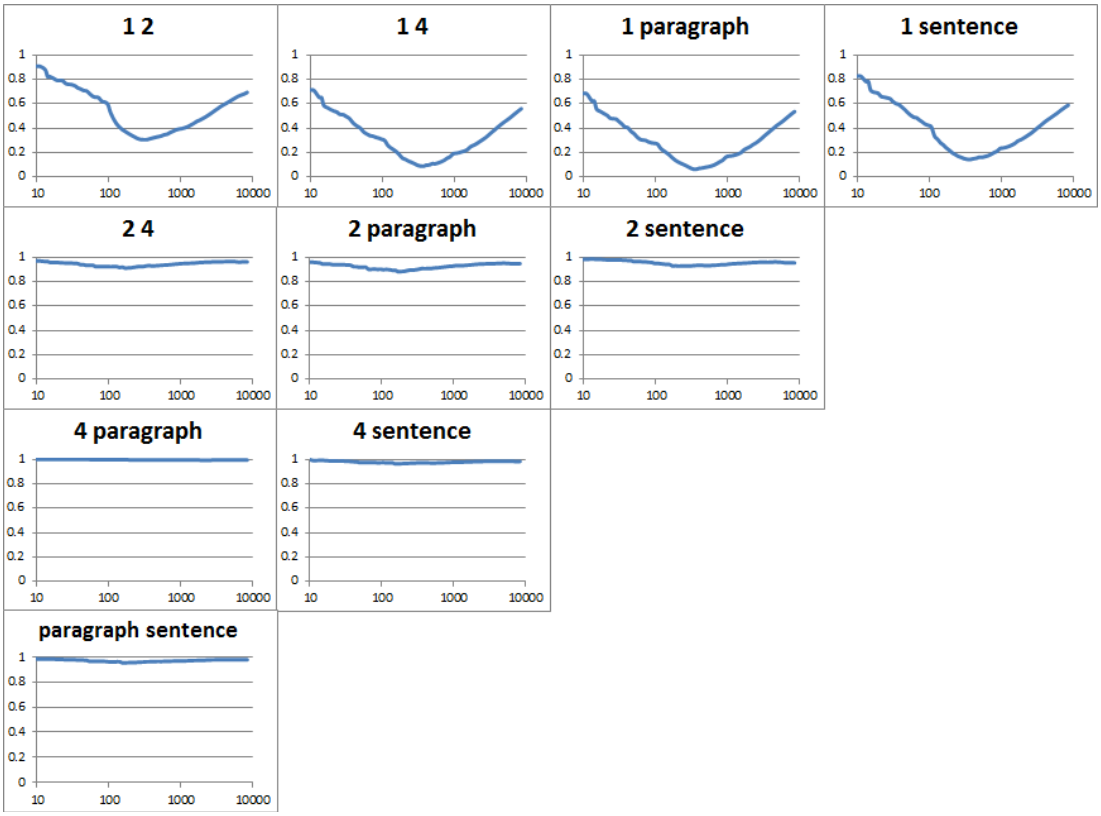

Figure 7. Comparison of different window sizes. The caption at the top of each graph indicates the window sizes being compared, and each graph shows similarity between two models as a function of the number of terms included in calculation of the Pearson coefficient between lists of terms most similar to Kant in each model (Y axis is from 0 to 1, X axis logarithmic 10 to 10,000).

Surprisingly, the window size had almost no effect on the similarity measure of the word 'Kant'. While the extremely small one-word window model varied slightly from the other window sizes, a two word window had an extremely high similarity with the longest paragraph length window selection that was tested.

7 Conclusions

With the exception of stemming, the parameter settings did not tend to have a large impact on the resulting models and therefore, the conclusions that can be drawn from these models. This is an extremely beneficial finding, as it indicates that these models can be used to understand large scale corpora without relying on something like a gold standard for optimization. In particular, dimensionality became of little relevance after around one or two thousand dimensions, and the gold standard model proposed in this paper completely negates the need for worrying about dimensionality under conditions in which it can be used. Window size, also had a negligible effect as long as a reasonable value was chosen (at least 2) with even the largest spread in variation (2 word compared to paragraph length) never having a correlation of less than .9. While strong and weak character lists may matter in multi-language corpora, the few occurrences (even if they are for high frequency words, such as particular philosophers) within a corpus like the SEP, also provide negligible variability.

Only the process of stemming provides a large scale modification of the results. While our procedures are by their nature incapable of addressing correctness (if that notion is even well-defined in this context), the experiments described here provide a cautionary tale for this one action. We suggest experimenters generate both forms of models and confirm that results obtained, pertain to both conditions, or that a valid argument can be used to validate a choice of an individual model. For example, if one is only interested in philosophers' beliefs, one may prefer to use a stemmer to get to the core meaning of the words whereas, when comparing writing between two philosophers, the particular words and relative relations would suggest not stemming the data set.

References

M. N. Jones and D. J. Mewhort (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological review, 114(1):1{37, 2007}*

Firth, J. R. (1957). A synopsis of linguistic theory, *1930-1955.*

Porter, M. (2009). Snowball: A language for stemming algorithms, 2001. URL http://snowball. tartarus. org/texts/introduction. html.

Kievit-Kylar, B., & Jones, M. N. (2012). Visualizing multiple word similarity measures. *Behavior research methods, 44(3), 656-674.*

Allen, C., & the InPhO Group (in press, 2013) "Cross-cutting categorization schemes in the digital humanities." Isis, forthcoming.