LSA of Conceptual Combination Stimuli

Rory Flemming

October 17, 2018

Word Pairs

```
# Import word pairs. The data is coded in the following way
# col 1- the first word of stimulus
# col 2- the second word of the stimulus
# col 3- 0 if 1st word is "Concrete", 1 if first word is "Abstract"
# col 4- order of stimulus presentation. 0 indicates the example pair in "training"
CC_stims <- read.csv(file='CC_stimuli_POA-5.csv',header=TRUE)
str(CC_stims)
```

```
## 'data.frame':    13 obs. of  4 variables:
##  $ word1 : Factor w/ 13 levels "chimney","frog",..: 3 2 6 5 11 10 12 8 7 13 ...
##  $ word2 : Factor w/ 13 levels "bliss","cactus",..: 8 9 4 11 10 6 7 2 1 13 ...
##  $ abs1st: int  1 0 1 1 0 0 1 1 0 1 ...
##  $ order : int  1 2 3 4 5 6 7 8 9 10 ...
```

Comparison Corpi

We will use three comparison corpi, for validation purposes:
* TASA - "Touchstone Applied Science Associates, Inc." A corpus of a broad set of topics and used to compile "The Educator's Word Frequency Guide." Built from >37.5k documents, and containing >92k different terms.
* EN_100k - The recommended space for computations in English. Absolutely massive (too big for me to venture downloading). ~2 Billion words, almost 5.4 million documents, with rows on 100k most frequent words. 5.4mill dimensions reduced to 300 via SVD.

```
load("TASA.rda")
TASA_mat = as.textmatrix(TASA)
```

LSA: Semantic Distance

The first two things I will look at are:
1) What are the magnitudes of the semantic distances between the first words and the second words of the stimulus pairs?
2) Are these cosine distances symmetric? (They should be)

```
# First, let's just get a read of the semantic distances of the pairs
for (i in 1:nrow(CC_stims)){ # For each CC stimulus
  # compute cosine distance of the words in the ith pair
  CC_stims$semantic_distance[i] = 1 - abs(Cosine(x=CC_stims$word1[i],
                                                 y=CC_stims$word2[i],tvectors=TASA_mat));
  # same computation, but switch their places
  CC_stims$semantic_distance2[i] = 1 - abs(Cosine(x=CC_stims$word2[i],
```

```
                                                    y=CC_stims$word1[i],tvectors=TASA_mat));
}
CC_stims # look at the results
```

```
##        word1   word2 abs1st order semantic_distance semantic_distance2
## 1    honesty  ladder      1     1         0.9713464          0.9713464
## 2       frog    luck      0     2         0.9849996          0.9849996
## 3      mercy    comb      1     3         0.9874850          0.9874850
## 4    justice  pillow      1     4         0.9761459          0.9761459
## 5      tiger paradox      0     5         0.9666753          0.9666753
## 6    thimble   glory      0     6         0.9546344          0.9546344
## 7  willpower    kite      1     7         0.9728395          0.9728395
## 8  reasoning  cactus      1     8         0.9837986          0.9837986
## 9     parrot   bliss      0     9         0.9479750          0.9479750
## 10    wisdom tractor      1    10         0.9820657          0.9820657
## 11    sponge purpose      0    11         0.9797330          0.9797330
## 12      hope   clock      1    12         0.8931780          0.8931780
## 13   chimney courage      0     0         0.9414438          0.9414438
```

The table shows us that the semantic distances of the stimuli range [0.893178, 0.987485], and that the distances, when computed by cosine distances are symmetric. Other distance metrics do allow for asymmetry. Really quick, I am going to download the EN_100k LSA space, and run this same analysis to see if we get similar results. Due to the size, more sophisticated analysis ought to be done on another machine, if it involves storing this data while manipulating it or other vaiables...

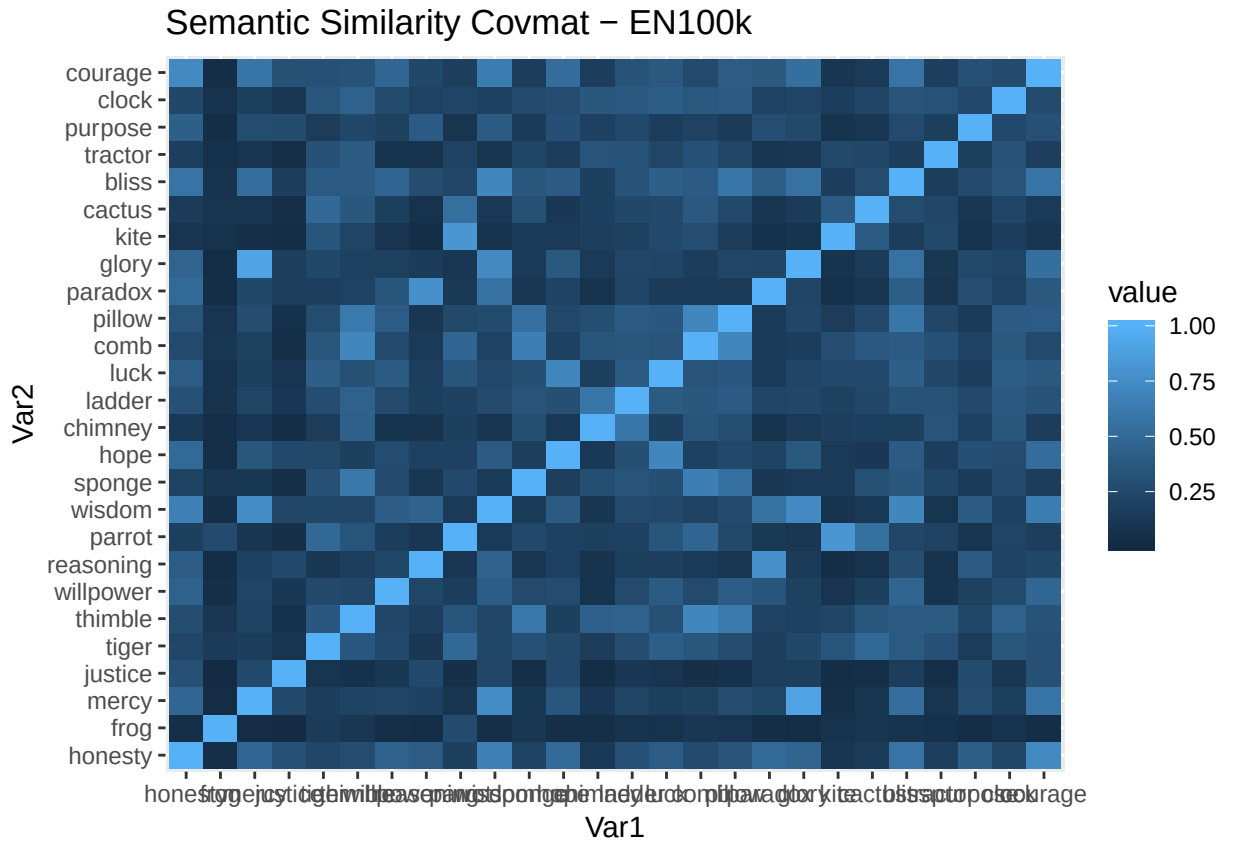
```
load('EN_100k_lsa.rda');
EN_100k = as.textmatrix(EN_100k_lsa);
for (i in 1:nrow(CC_stims)){ # For each CC stimulus
  # compute cosine distance of the words in the ith pair
  CC_stims$semantic_distance2[i] = 1 - abs(Cosine(x=CC_stims$word2[i],
                                                  y=CC_stims$word1[i],tvectors=EN_100k));
}
CC_stims
```

```
##        word1   word2 abs1st order semantic_distance semantic_distance2
## 1    honesty  ladder      1     1         0.9713464          0.6909368
## 2       frog    luck      0     2         0.9849996          0.9331300
## 3      mercy    comb      1     3         0.9874850          0.8122425
## 4    justice  pillow      1     4         0.9761459          0.9387678
## 5      tiger paradox      0     5         0.9666753          0.8277675
## 6    thimble   glory      0     6         0.9546344          0.8085583
## 7  willpower    kite      1     7         0.9728395          0.9120745
## 8  reasoning  cactus      1     8         0.9837986          0.9328778
## 9     parrot   bliss      0     9         0.9479750          0.7657528
## 10    wisdom tractor      1    10         0.9820657          0.9097992
## 11    sponge purpose      0    11         0.9797330          0.8549207
## 12      hope   clock      1    12         0.8931780          0.7198179
## 13   chimney courage      0     0         0.9414438          0.8411402
```

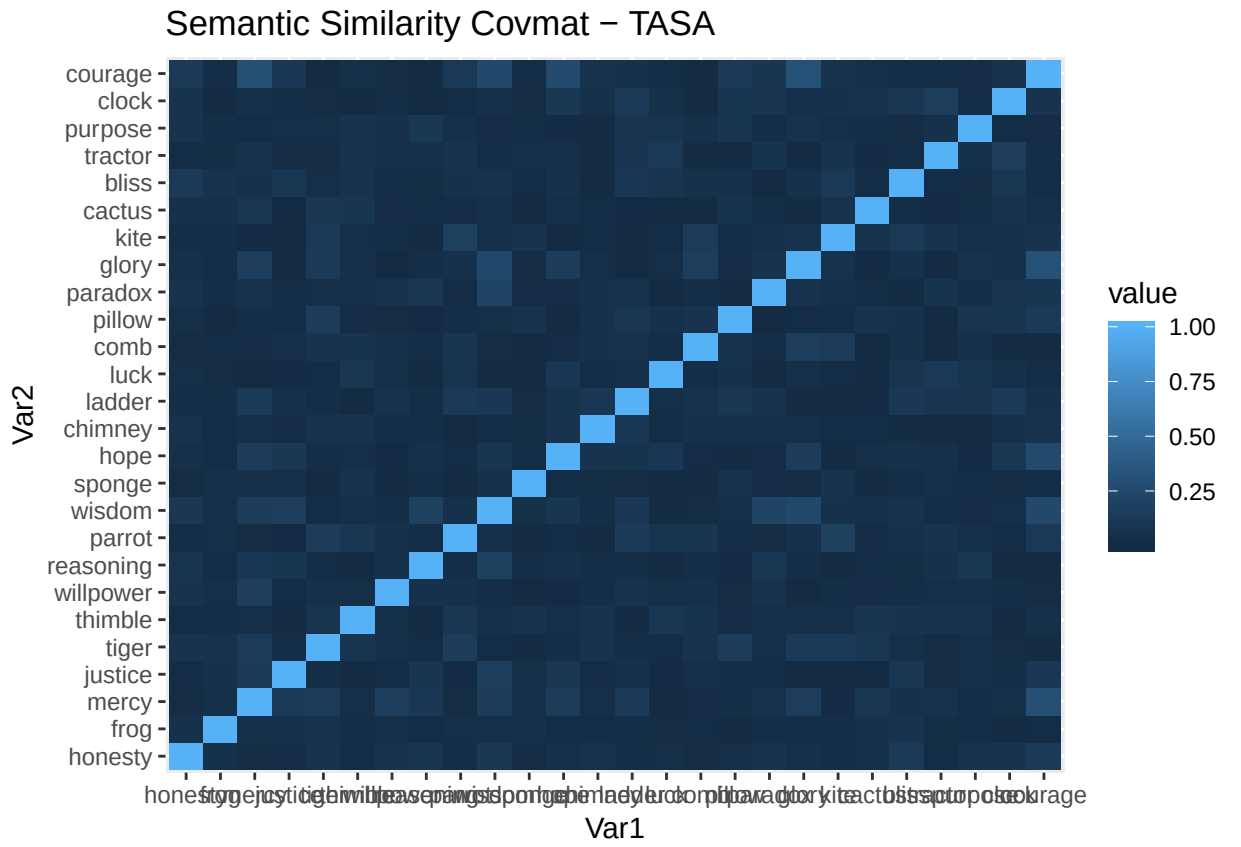
Woah!! We actually get way different, and richer results. There is more variability at the very lease. Maybe the TASA corpus is not appropriate for our word content. I would be more inclined to "trust" the EN_100k since it is so much larger and more general. Still, it is a tough quest to decide on an appropriate LSA

space... Before tossing this out, let's have a look at the results just from these two and see if there is some correlation between them...
What I'm going to try to do next is to create the covariance matrix for each of the stimulus words in each of the spaces, then compare them to one another. The point of doing this would be to see if we can correctly identify "concrete" vs "abstract" words in the stimulus set.

```
# Separate out the words and their label (Abs, concrete)
words = c(as.character(CC_stims$word1),as.character(CC_stims$word2));
labels = c(CC_stims$abs1st,(1-abs(CC_stims$abs1st)));

# Compute semantic simmilarity on all of the words
words_simmat_EN_100k = abs(multicos(words,words,EN_100k));
word_simmat_TASA = abs(multicos(words,words,TASA_mat));

melted_EN100k = melt(words_simmat_EN_100k);
melted_TASA = melt(word_simmat_TASA);

ggplot(melted_EN100k, aes(x=Var1,y=Var2,fill=value)) +
  geom_tile() + labs(title="Semantic Similarity Covmat - EN100k")
```




```
ggplot(melted_TASA, aes(x=Var1,y=Var2,fill=value)) +
  geom_tile() + labs(title="Semantic Similarity Covmat - TASA")
```



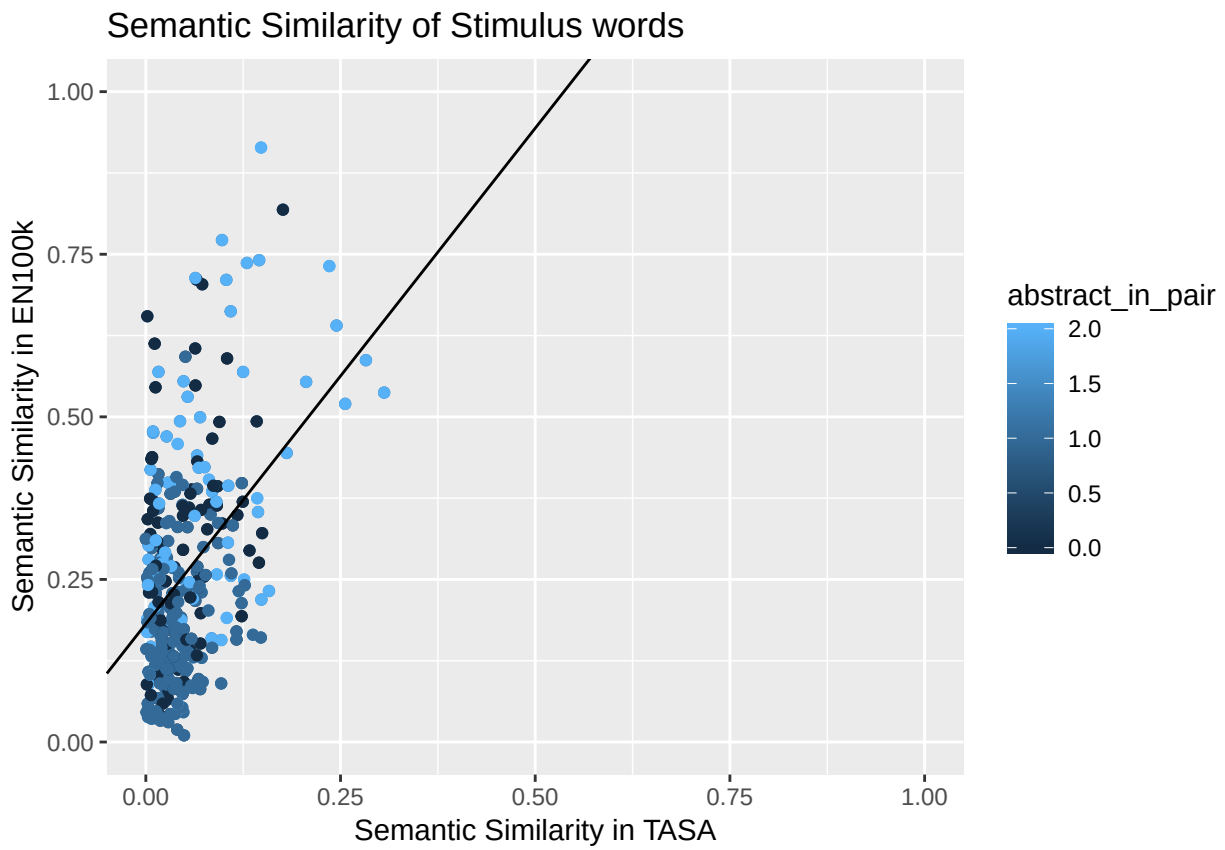
Now, these huge matrices are a bit hard to interpret... Now, how can we look at their simmilarity? I'm going to start super simple and just ask: do the two LSA spaces show a consistent pattern of correlation? In other words, is there a positive relationship between the covariances calculated in TASA and EN_100k? Here, I fit a simple lienar model:

Let's look at the results:

```
# Report the coefficients
TASA2EN100k$coefficients
```

```
## (Intercept)           x
##   0.1816351   1.5243273
```

```
# Color mask
abstract_labels = c(as.integer(CC_stims$abs1st),as.integer(!CC_stims$abs1st));#abs/conc for every word
abstract_mat = outer(abstract_labels,abstract_labels,FUN="+"); # create matrix of combinations
melted_abs = melt(abstract_mat); #melt to fit format
melted_abs = as.array(melted_abs$value); # only use the value column (contains the actual labels)
non_diag_mask = 1:length(melted_abs); non_diag_mask = non_diag_mask[(non_diag_mask%%(2*length(CC_stims$a
abstract_in_pair = melted_abs[non_diag_mask]# remove the diagonal values


#Plot model fit
ggplot(data=NULL,aes(x=x,y=y,color=abstract_in_pair)) +
  geom_point() + labs(x="Semantic Similarity in TASA",y="Semantic Similarity in EN100k", title="Semanti
  geom_abline(slope=TASA2EN100k$coefficients[2],intercept = TASA2EN100k$coefficients[1]) + lims(x=c(0,1
```



Cool!! There is possibly something interesting here. If we were to get the mean similarities for concrete-abstract pairs we would likely see that they are smallest, then concrete pairs, and finally abstract pairs. Next let's look at just our 13 pairs, and compare them to a set of test words that we have. These test words were selected because they have certain relationships that *could* inform our intuition regarding whether the relative distances of our stimuli could be meaningful.

```
test_pairs <- read.csv(file='CC_stim_comparisons.csv',header=TRUE)
str(test_pairs)
```

```
## 'data.frame':    9 obs. of  5 variables:
##  $ word1        : Factor w/ 9 levels "black","coconut",..: 7 4 6 1 5 8 3 9 2
##  $ word2        : Factor w/ 9 levels "boot","computer",..: 3 2 1 9 4 7 5 6 8
##  $ abstractpair : int  -1 -1 -1 0 1 1 1 -1 -1
##  $ concreteness1: num  4.9 4.88 4.97 3.76 2.5 1.96 2.6 NaN NaN
##  $ concretenes2 : num  4.71 4.89 5 3.89 1.97 3.11 1.96 NaN NaN
```

These are our comparison pairs. Included for some of them is whether they are concrete or abstract pairs, and what the concreteness rating of each word is (only for some pairs). Let's get their semantic distances first, then we will append them to the stimulus pairs, and finally, we will add some other metadata that might be useful.

```
for (i in 1:nrow(test_pairs)){
  test_pairs$semantic_distance[i]  = 1 - abs(Cosine(x=test_pairs$word1[i],
                                          y=test_pairs$word2[i],tvectors=TASA_mat));
    test_pairs$semantic_distance2[i] = 1 - abs(Cosine(x=test_pairs$word1[i],
                                          y=test_pairs$word2[i],tvectors=EN_100k));
}
```

```
## Note: x converted to character

## Note: y converted to character

## Note: x converted to character

## Note: y converted to character

## Note: x converted to character

## Note: y converted to character

## Note: x converted to character

## Note: y converted to character

## Note: x converted to character

## Note: y converted to character

## Note: x converted to character

## Note: y converted to character

## Note: x converted to character

## Note: y converted to character

## Note: x converted to character

## Note: y converted to character

## Note: x converted to character

## Note: y converted to character

## Note: x converted to character

## Note: y converted to character

## Note: x converted to character

## Note: y converted to character

## Note: x converted to character

## Note: y converted to character
```

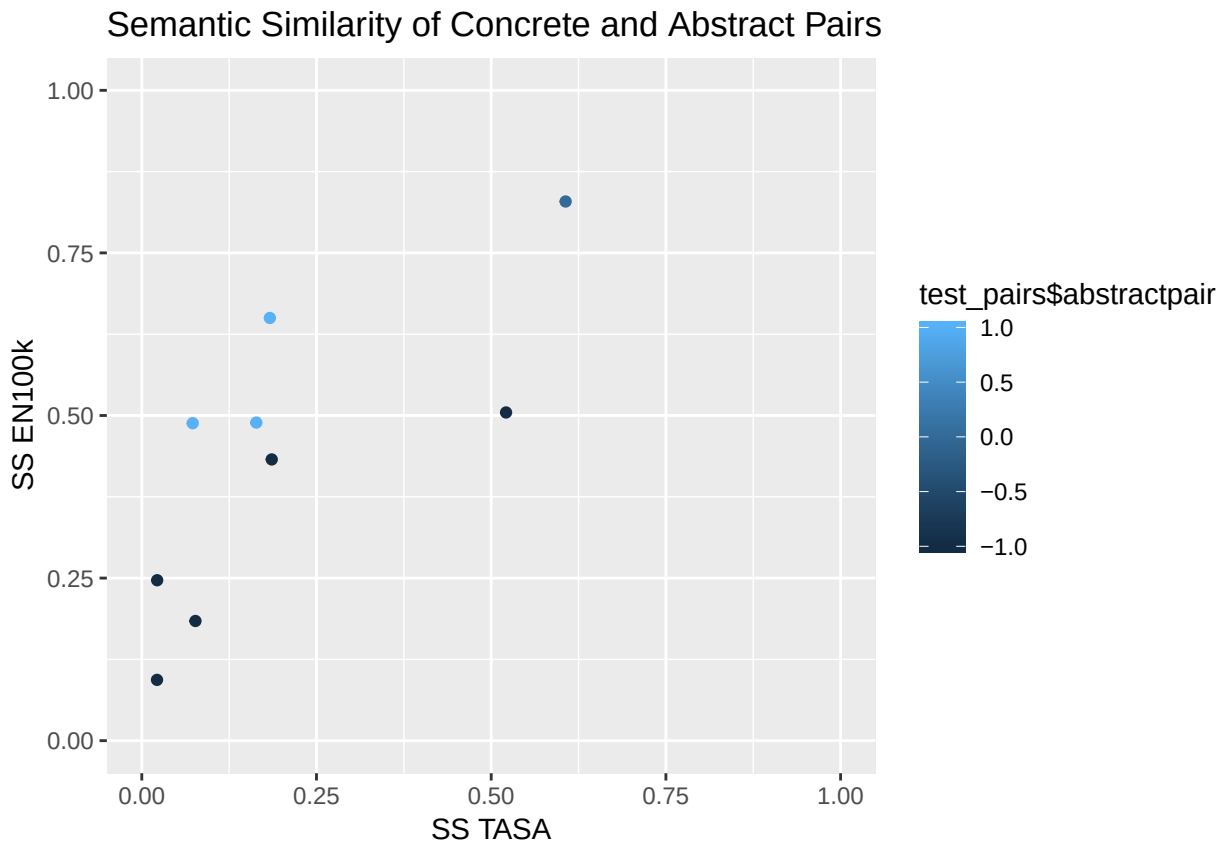
```
## Note: x converted to character

## Note: y converted to character

## Note: x converted to character

## Note: y converted to character

## Note: x converted to character

## Note: y converted to character

## Note: x converted to character

## Note: y converted to character

## Note: x converted to character

## Note: y converted to character

## Note: x converted to character

## Note: y converted to character
```

```
test_pairs$synonym = c(1,0,1,-1,-1,-1,1,0,0);
test_pairs$utility = c(1,1,1,0,0,0,1,0,0);
test_pairs$context = c(0,1,1,0,0,0,0,0,0);
test_pairs
```

```
##       word1      word2 abstractpair concreteness1 concretenes2
## 1      sofa      couch           -1          4.90         4.71
## 2    pencil   computer           -1          4.88         4.89
## 3      shoe       boot           -1          4.97         5.00
## 4     black      white            0          3.76         3.89
## 5      real       fake            1          2.50         1.97
## 6     truth        lie            1          1.96         3.11
## 7 happiness   gladness            1          2.60         1.96
## 8 waterfall     jacket           -1           NaN          NaN
## 9   coconut skateboard           -1           NaN          NaN
##   semantic_distance semantic_distance2 synonym utility context
## 1         0.4787473          0.4953456       1       1       0
## 2         0.9780223          0.7532734       0       1       1
## 3         0.8140576          0.5675390       1       1       1
## 4         0.3935058          0.1709297      -1       0       0
## 5         0.8361267          0.5108650      -1       0       0
## 6         0.8167338          0.3500230      -1       0       0
## 7         0.9271988          0.5118558       1       1       0
## 8         0.9232569          0.8159607       0       0       0
## 9         0.9782360          0.9064793       0       0       0
```

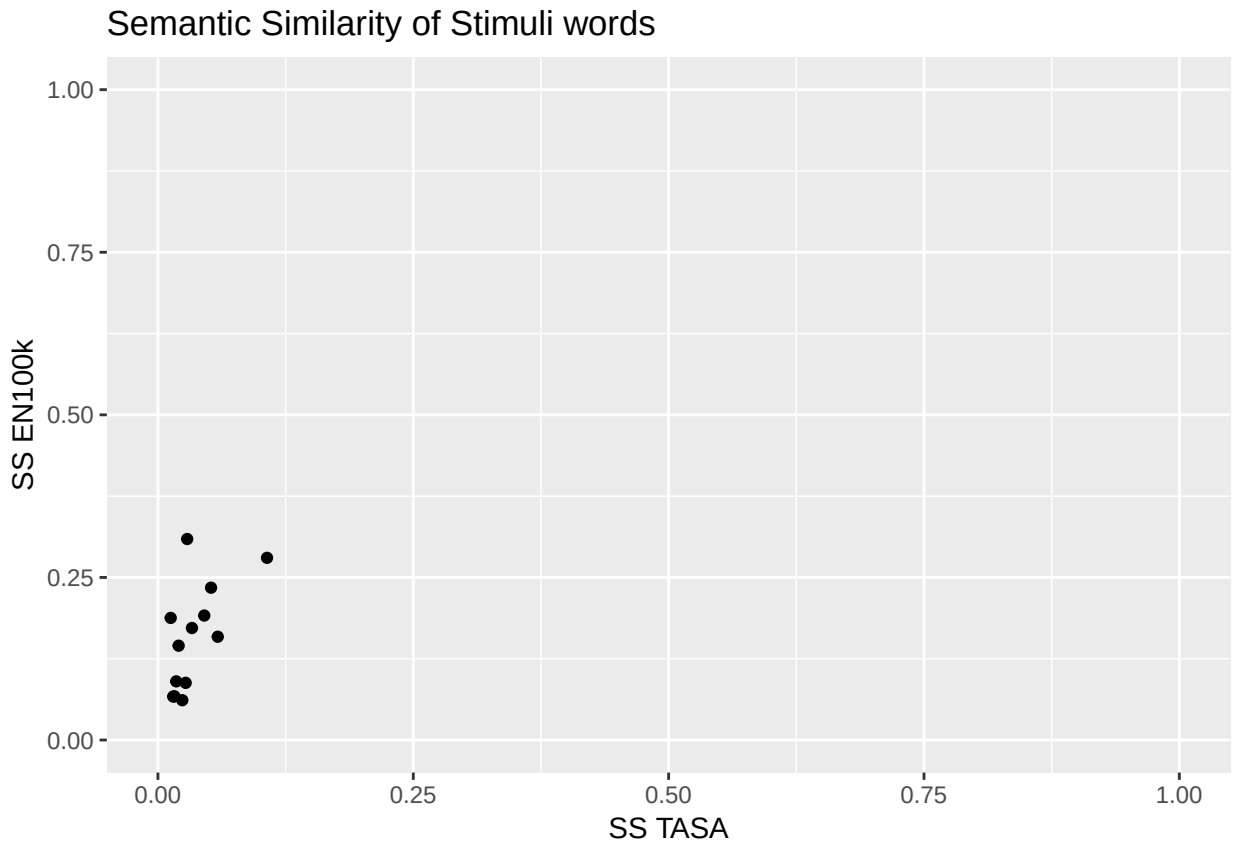
First, let's look at the comparison of abstract-concrete pairs (and "mid-level" b&w pair):

```
ggplot(data=NULL,aes(x= 1-test_pairs$semantic_distance,
                     y=1-test_pairs$semantic_distance2,color=test_pairs$abstractpair)) + geom_point()
```



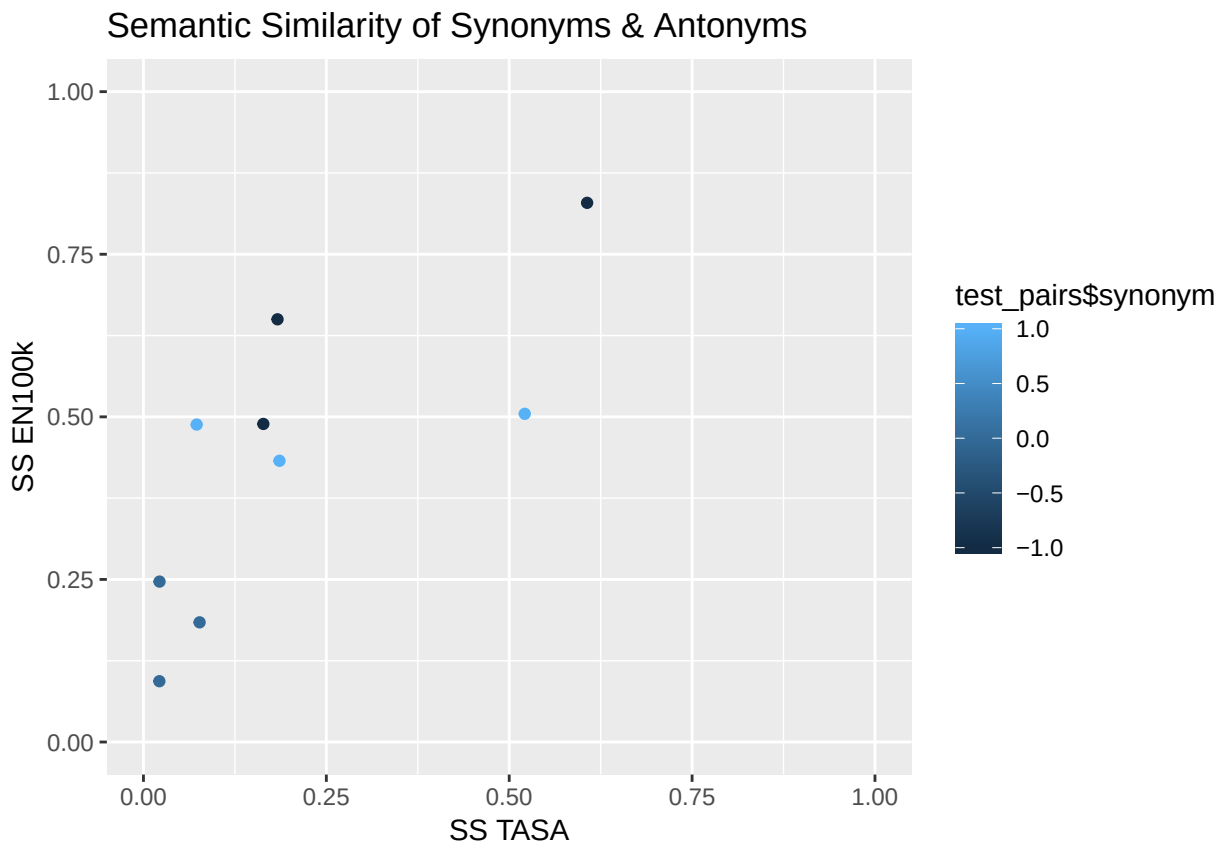
Let's look at our stimuli...

```
ggplot(data=NULL,aes(x=1-CC_stims$semantic_distance,y=1-CC_stims$semantic_distance2)) +geom_point() + la
```



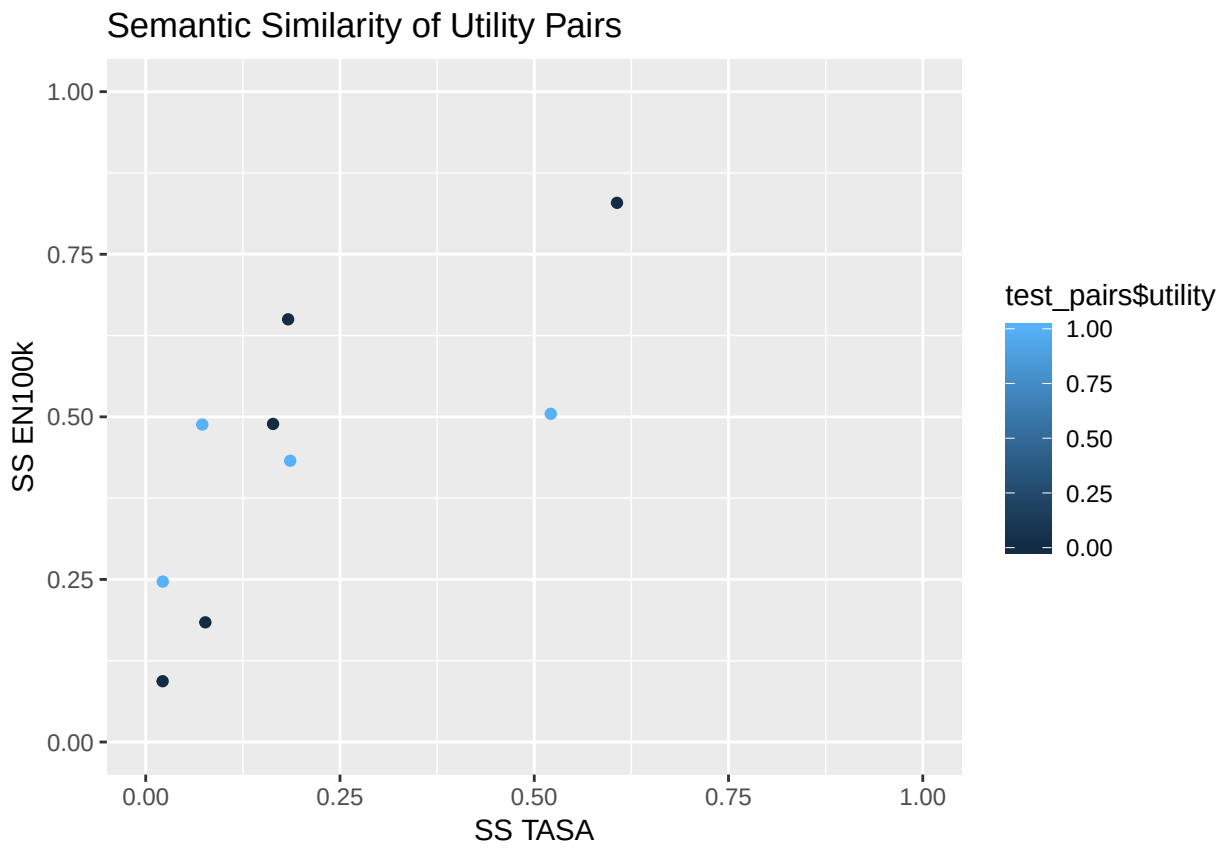
Let's look at synonyms vs antonyms and the pairs with no relations.

```
ggplot(data=NULL,aes(x= 1-test_pairs$semantic_distance,
                     y=1-test_pairs$semantic_distance2,color=test_pairs$synonym)) + geom_point() + la
```



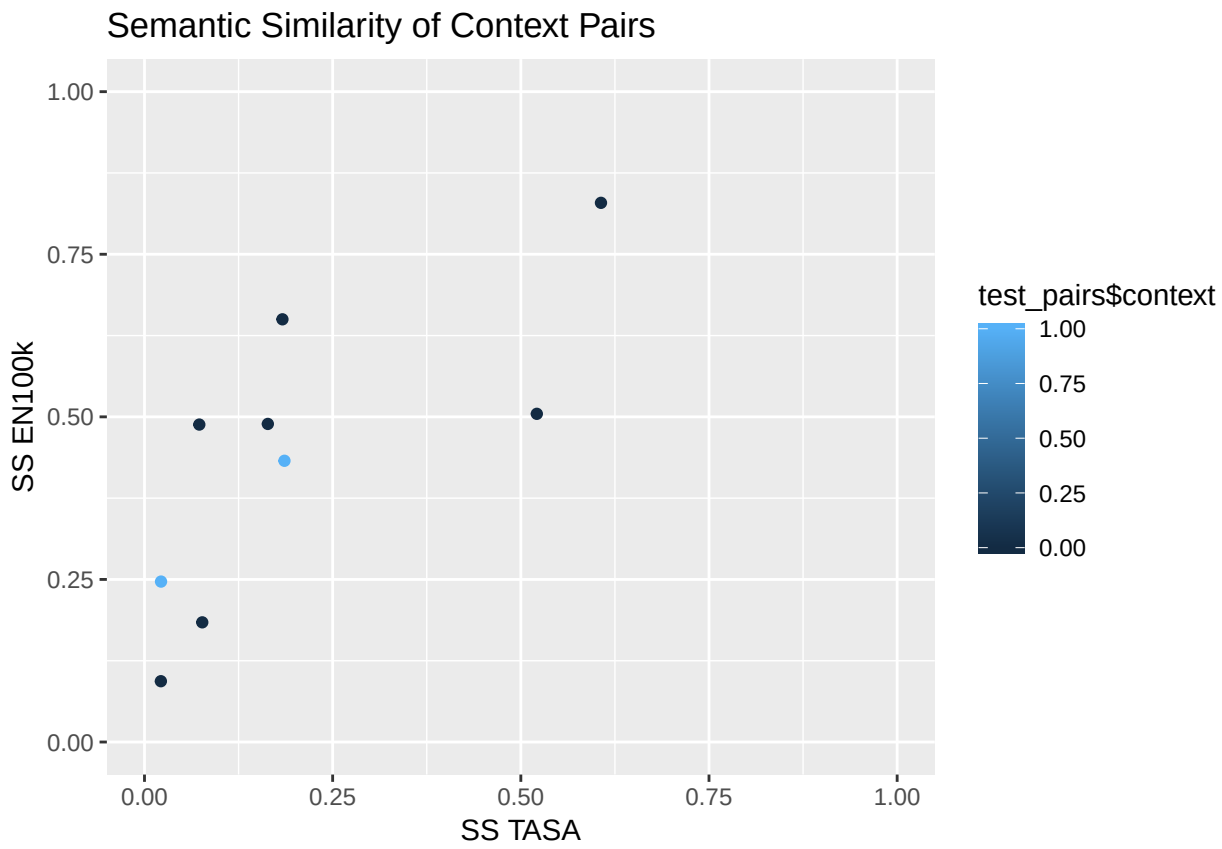
Let's look at the pairs which might have similar utility/context

```
ggplot(data=NULL,aes(x= 1-test_pairs$semantic_distance,
                     y=1-test_pairs$semantic_distance2,color=test_pairs$utility)) + geom_point() + labs
```



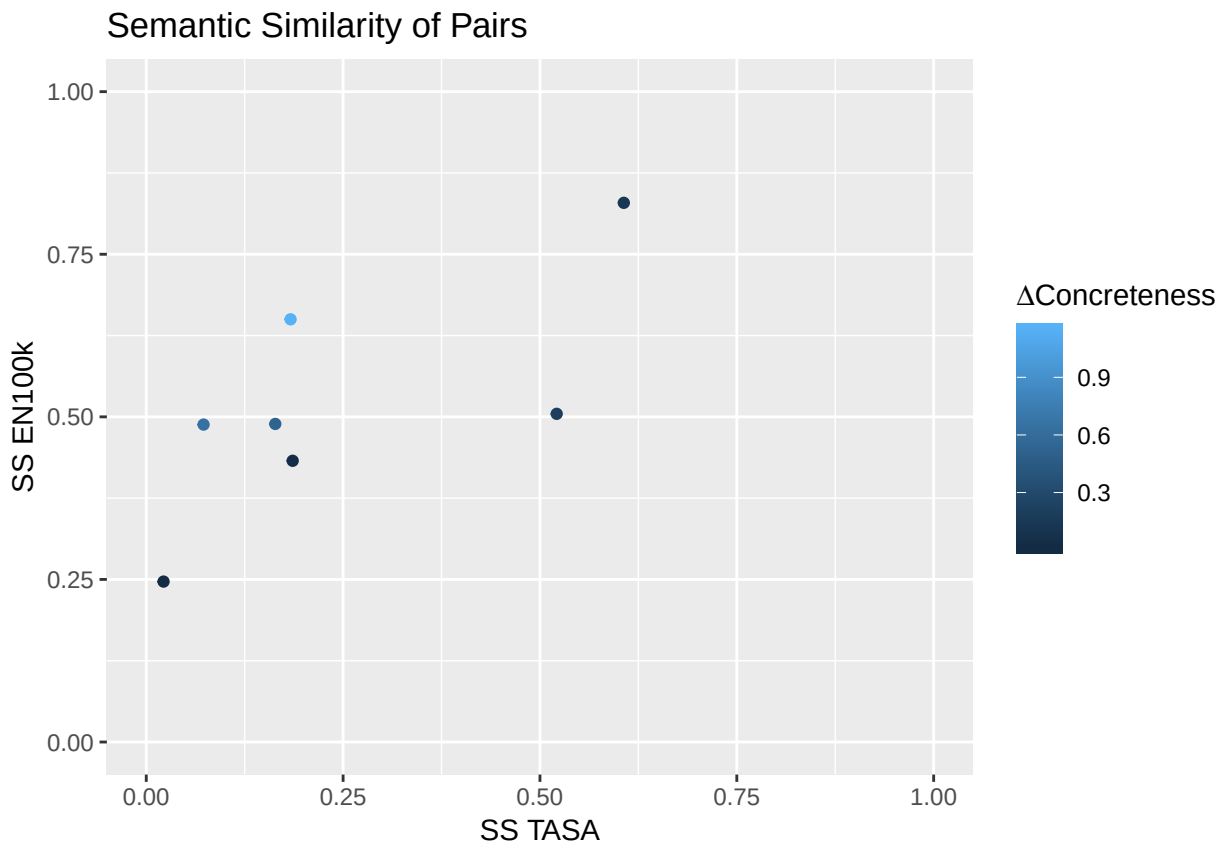
And finally context:

```
ggplot(data=NULL,aes(x= 1-test_pairs$semantic_distance,
                     y=1-test_pairs$semantic_distance2,color=test_pairs$context)) + geom_point() + labs(
```



What about the distance in concreteness rating?

```
test_pairs$concrete_dif = abs(test_pairs$concretenes2 - test_pairs$concreteness1)
ggplot(data=NULL,aes(x= (1-test_pairs$semantic_distance[1:7]),
                     y=(1-test_pairs$semantic_distance2[1:7]),
                    color= test_pairs$concrete_dif[1:7]))+ geom_point() + labs(x="SS TASA",y="SS EN100
```



A note: with so few exemplars it's really hard to say anything concrete (haha) about how these constructs work in this space. We know that 1) our stimuli have low semantic similarity (high semantic difference) compared to these example sets. Additionally, randomly select abstract-concrete mismatch pairs have smaller semantic similarity than matched pairs of either type (eye-ball test, we could run the statistics to actually test). It's uncertain whether abstract or concrete pairs have comparable semantic similarities, though from our earlier graphic, it appears that maye abstract pairs could be biased more similar than concrete, as "abstractness" itself is a viable point of similarity, more narrowly bounding than "concrete." One thing about these "analyses" is that they are rather high level, maybe not at the granularity we may desire. LSA uses a term x document matrix, which may be too far removed to get at the things we are interested in. It could be possible that using wordNet or a sentence2vec, paragraph2vec, or another, smaller moving window could provide a more informative and granular perspective.