Fall 12-18-2014

An empirical study of semantic similarity in WordNet and Word2Vec

Abram Handler
ahandler@uno.edu

An empirical study of semantic similarity in WordNet and Word2Vec

A Thesis

Submitted to the Graduate Faculty of the
University of New Orleans
in partial fulfillment of the
requirements for the degree of

Master of Science
in
Computer Science

by

Abram Handler

B.A. Columbia University, 2007 (Hons.)

December 2014

Contents

Abstract

This thesis performs an empirical analysis of Word2Vec by comparing its output to WordNet, a well-known, human-curated lexical database. It finds that Word2Vec tends to uncover more of certain types of semantic relations than others – with Word2Vec returning more hypernyms, synonomyns and hyponyms than hyponyms or holonyms. It also shows the probability that neighbors separated by a given cosine distance in Word2Vec are semantically related in WordNet. This result both adds to our understanding of the still-unknown Word2Vec and helps to benchmark new semantic tools built from word vectors.

Word2Vec, Natural Language Processing, WordNet, Distributional Semantics

1 Introduction

Word2Vec is a new unsupervised system for determining the semantic distance between words. For instance, after learning from billions of web pages, Word2Vec reports that the words *Chinese river* are semantically close to the word *Yangtze*. [1] Such results have attracted lots of recent attention: over 100 researchers have cited Word2Vec since its publication in 2013. Yet certain aspects of the system's output are poorly understood. In particular:

1. Word2Vec does not label particular semantic relationships between words – like the synonomy between *cold* and *chilly* or the meronomy between *wheel* and *car*. Instead, it assigns a number between 0 and 1, indicating the semantic distance between two words[1]. However, as Word2Vec's creators note "there can be many different types of similarities." [2] This opens a question: what sorts of semantic similarities does Word2Vec uncover?

2. Word2Vec can generate ranked lists showing which words are closer and which words are further way in a semantic model. For example, Word2Vec says that *grandmaster* is 3rd from the word *chess*, while *Muay Thai kickboxing* is 997th. [2] What is the probability that two words that are some distance apart in Word2Vec stand in some formal specific semantic relationship?

This study seeks to answer such questions by comparing Word2Vec's output with WordNet – a large, human-curated "lexical database" [3] which is the most-frequently cited "lexiographic resource" [4] in English.

Such effort has several motivations. First, the study simply gives clearer knowledge of Word2Vec, which is still not well understood. Second, as researchers and practitioners build semantic tools from Word2Vec, they will inevitably turn to WordNet to evaluate their applications. Rei et. all [5] have already tried using WordNet to benchmark their Word2Vec-based hyponym detector. Accurately benchmarking such tools requires a clear understanding of the relationship between the two semantic systems. For instance, to evaluate Rei's study we must ask: what is the probability that Word2Vec will

[1] For instance, a Word2Vec model trained on the Google news corpus returns a semantic distance of .390 between *truck* and *tire* and a semantic distance of .168 between *truck* and *chicken*

[2] Word2Vec model trained on Google news corpus

return a holonym from a random word at a particular semantic distance? This study establishes such a baseline for further research.

2 Related Work

2.1 Geometry and Meaning

Computers are much better than humans at certain tasks, such as searching large lists or solving complex equations. However, researchers and programmers still struggle with the highly nuanced, contextually-dependent work of understanding a word's meaning. While automatic translation services might approximate some of our human intuitions about the meaning of a word or phrase, replicating all of the intricacy of natural language semantics remains an unsolved problem in computer science.

Efforts thus far have presumed a so-called distributional theory of semantics, which hypothesizes that those words which are distributed the same way in text or speech will have similar meanings.

According to the distributional theory, the words *keyboard* and *piano* might occur together frequently in text because they refer to related things in the world. This semantic approach is often distilled into a quip from the linguist JR Firth: "a word is characterized by the company it keeps." [6] [3]

Translating the linguistic insight into algorithmic formality often entails joining all of the words and all of the documents in a corpus to form the rows and the columns of a large matrix – which is then condensed into a smaller matrix of more manageable size. When words and phrases are projected into this semantic space, those words that have similar meanings are closer together. Those that have less similar meanings are farther apart.

As Dominic Widdows points out – such algorithms form an unlikely connection between geometry (the mathematical study of space and objects in space) and semantics (the way a word refers to an object in the world). In other words: an unexpected link between geometry and meaning. *Geometry and Meaning* [4]. Word2Vec is a new method, but falls squarely within this decades-long tradition of research.

[3] Understanding natural language programmatically slides into philosophy and linguistics, where theorists and have debated how words gain their meanings for millenia [7]. We do not dig into the details of such debates here – but instead take the distributional theory as given.

2.2 Word2Vec

Word2Vec follows very much within in the geometric methods detailed in section 2.1. The algorithm uses complex, multi-level neural networks to project words into a semantic space, which can then be used to determine semantic distance or semantic proximity. The network is trained by giving positive feedback when words appear together in context and giving negative feedback when words are randomly swapped into other contexts. The output is a space filled with vector representations of words. Word vectors that are semantically closer together are more closely related than word vectors that are farther apart.

This approach has garnered lots of attention and enthusiasm. Researchers have tried using Word2Vec to find the meaning of a word in context [8], to automatically determine human attitudes in text [9] and even to ascertain political ideology [10]. Yet no empirical studies have yet attempted to systematically analyze output from Word2Vec in terms of the classical tool, WordNet.

The gap is notable, in part, because researchers have begun to evaluate new semantic tools built on top of Word2Vec by using the human-curated WordNet, which remains the most-precise method for determining semantic relationships with a computer.

Rei et. all's "Looking for Hyponyms in Vector Space" serves as an important example. [5] The researchers first use Word2Vec to find words with a particular semantic relationship (hyponomy) – then look to WordNet to evaluate their method. Yet they presuppose certain relationships between WordNet and Word2Vec without providing any empirical justification – writing that "the most likely candidates for a high cosine similarity are synonyms, antonyms, hypernyms and homonyms". In section 4.4 we show that this is not the case. Word2Vec does not in fact return these semantic relations equally.

Clearer understanding of how WordNet and Word2Vec are related will yield much precise evaluations. After all, both the linguistic mechanisms and the exact output of the Word2Vec system are still a bit mysterious. One popular explanation of the system concludes: "Why does this produce good word representations? Good question. We don't really know." [11]

2.3 WordNet

WordNet has been a part of natural language processing for decades – beginning at Princeton University in 1986. The system has its own logic and jargon, all built around the fundamental building block [12] of the "synonymous set" (or synset) – an unordered collection of "cognitively synonymous words and phrases." [13] Synsets are linked by relations, with particular relations linking particular words with particular parts of speech.

Thus, where Word2Vec represents words as vectors, WordNet models language with a large graph – with semantically similar words (called "synsets") serving as the nodes and semantic relationships (such as the meronomy between *tire* and *car*) serving as the edges. [4]

Five specific WordNet relationships concern us here:

- **Synonomy**. WordNet can identify synonyms, words with the same meaning. For instance: *roof* and *ceiling*.
- **Hypernomy**. A word that is more general than some other word is said to be its hypernym. *Language* is a hypernym of *French*
- **Hyponomy**. A word that is more specific than some other word is said to be its hyponym. *French* is a hyponym of *language*
- **Meronomy**. A word that is a part of some other word is called a meronym. *Bedroom* is a meronym of *house*.
- **Holonomy**. If B contains A, B is a holonym of A. *Japan* is a holonym of *Fuji* because Fuji is in Japan.

Note that these are relationships between synsets, not between words. A word is associated with one or more synsets. A synset has a semantic relationship with other synsets. We explain our exact definition of semantic relation in the section 3.

3 Method

If word vectors represent semantic similarity, we might expect a that two words that are a certain distance apart in Word2Vec have some semantic

[4] Because "the majority of the WordNet's relations connect words from the same part of speech (POS) ... WordNet really consists of four sub-nets, one each for nouns, verbs, adjectives and adverbs, with few cross-POS pointers."[3]

relationship in WordNet. We investigate with the following experiment.

We begin with Google's word model, trained on 100 billion words in the Google news corpus [1]. Then we select the set of 41600 unique words from the Reuters news corpus and, for each, we search for the closet 200 neighbors in the Word2Vec model. We then search WordNet for any semantic relationship between the original word and its neighbor – concerning ourselves only with the semantic relations listed in section 2.3.

For instance, our experiment might extract the word *introduction* from the Reuters corpus and extract its 200 closest neighbors in Word2Vec. It might then find that k-th neighbor in Word2Vec is the word *initiation*. It would then look up synsets for each word in WordNet see if the two words have any synsets in common. In this case, it would determine that the two words are synonyms because their synsets overlap with the synset 'initiation.n.01' – defined as 'a formal entry into an organization or position or office'.

Note that this means that we use a binary measure to determine semantic relatedness in WordNet: if there is any overlap between the relevant synsets, we count the relation. If there is zero overlap between the synsets, we do not count the relation. Potential problems with such a binary measure are considered in section 4.6.

The process is very similar for other relations but the details warrant mention. Words can have multiple associated synsets in WordNet. When we search for holonyms, meronyms, hyponyms and hypernyms, we use a very wide measure of relatedness. For each synset associated with a word: we create a union of all associated relations from all synsets. If any synset from this union intersects with the synsets of the original word, we count the relation. Again, potential problems with such a measure are considered in section 4.6.

WordNet synsets are associated with a part of speech, but we do not consider them here. We discuss this further in section 5.

3.1 Implementation Details

There are a few important details surrounding the implementation of our experiment.

1. We allow two words to stand in multiple semantic relationships. For instance, a word is permitted to be both a hyponym and a holonym, if

Algorithm 1 Calculate the cosine distances associated with different semantic relations

```
W ← words
for i = 0 to len(W) do
  N ← word2vec − neighbors(200)
  for j = 0 to len(W) do
    if W[i] in wordnet then
      if samestem(W[i], N[j]) then
        record same stem and relation
      else
        if synononms(W[i], N[j]) then
          record cosine distance and relation
        end if
        if meronyms(W[i], N[j]) then
          record cosine distance and relation
        end if
        if hypernyms(W[i], N[j]) then
          record cosine distance and relation
        end if
        if hyponyms(W[i], N[j]) then
          record cosine distance and relation
        end if
        if holonyms(W[i], N[j]) then
          record cosine distance and relation
        end if
      end if
    else
      record not in WordNet
    end if
  end for
end for
```

so labeled in WordNet.

2. We also keep track of words that do not appear in WordNet and words that have the same stem – like *nearing* and *nears*. In these two cases, we do not search for the semantic relationships in WordNet because the relationship is already known.
3. We do not consider antonyms as Wordnet defines antonyms between words (not between synsets). All other relations are defined between synset, so antonyms are not an equal comparison.
4. We access WordNet 3.0 and the Reuters-21578 [14] corpus with the Python linguistics toolkit, NLTK [15].
5. We access WordNet 3.0 and the Reuters-21578 [14] corpus with the Python linguistics toolkit, NLTK
6. We access the Google news model via the popular python wrapper, Gensim [16].
7. We use NLTK's snowball stemming tool to find words with the same stem.
8. We ignore English stopwords, as defined by NLTK.
9. We use cosine distance as our measure of distance between vectors.
10. It is possible to train Word2Vec on any corpus of text – but we do not do so here. Instead we use the Word2Vec model that Google trained on the 100 billion word tokens in its Google news corpus. [1]

3.2 Processing results

The experiment detailed in section 3 lists the cosine distance between words. The section 4 unpacks its findings. However, translating the output from the section 3 into the results in 4 requires an intermediate step: discovering how many of each relation maybe found within a given semantic distance. We do so by dividing 1 into 1000 equal parts using the numpy linspace method. Then, we loop through each of these parts – and, for each, count how many of each type of relation are discovered beneath a particular threshold. This counting method generates many of our graphical results.

3.3 Adjusted counts

Some relations in WordNet have more associated synsets than others. Thus our experiment might be said to measure differences in associated synsets in WordNet instead of output from Word2Vec. We address this deficiency by generating adjusted counts that show the probability of each relation – if all relations contained equally many synsets. We do this as follows. First we randomly sample 10,000 words from the Reuters corpus and determine the average number of associated synonyms, meronyms, holoynms, hypernyms and hyponyms. Then we average these averages to get an overall average number of associated synsets. To generate an adjusted count for some relation, we multiply the raw count by the overall average divided by the average for synset. Thus, if some synset has twice as many associated words than average its count will be halved. If some synset has half as many as the overall average, its count will be doubled.

4 Results

4.1 Raw numbers: A high-level overview

Our first result is very clear: we show what sorts of relations are returned by Word2Vec.

Relation	Count
Holonym	1453
Meronym	2561
Hyponym	25620
Synonym	37107
Hypernym	42908

Table 1: Word2Vec captures far more of certain relations than others: favoring synonyms, hyponyms and hypernyms ahead of holonyms and meronyms across the Reuters corpus

We find that Word2Vec favors synonyms and hypernyms and hyponyms ahead of meronyms and holonyms by an order of magnitude. These large differences across the entire corpus seem to indicate that Word2Vec picks up certain relationships ahead of others.

We find that by far the largest category are those results that do not appear at all in WordNet. From our experiment, we find 1167354 such words

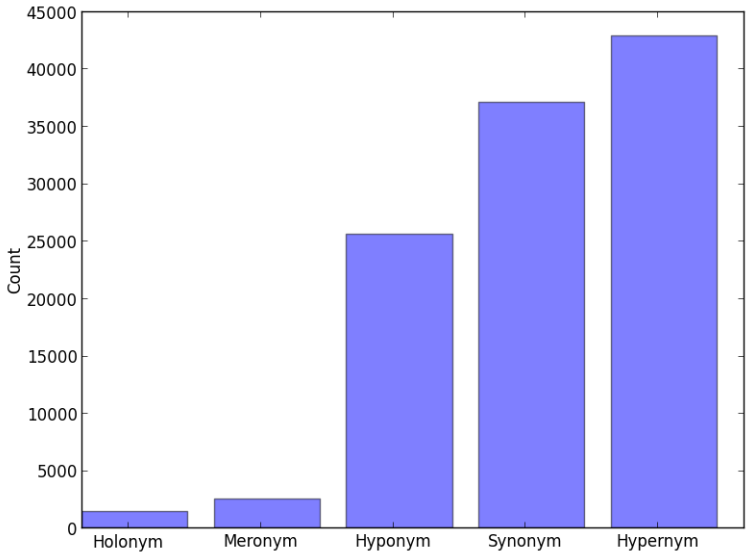

Figure 1: Word2Vec more synonyms, hyponyms and hypernyms than holonyms and meronyms

from Word2Vec – far too many to show in the chart above. This is because the model trained on the Google news corpus with Word2Vec is much, much larger than all of WordNet. Where WordNet 3 contains around 118,000 synsets [12], the Word2Vec model in this experiments was trained on 100 billion words [1] of news text. This means that most of the semantically similar words cannot be looked up in WordNet. For instance, *Rohto Pharmaceutical* is not in WordNet: it's a big corporation, but not a household name in the United States. Thus WordNet has no way of determining its semantic relationship to the word *industries*.[5] It is not known how well the WordNet relations represent the mass of 'semantically similar' words in Word2Vec. We consider this in section 5.

4.2 Frequency by cosine distance

Cosine distance is a measure of the distance between the angle of two vectors. We analyzed the relative frequencies of different relations at different cosine distances and found that relations were not distributed uniformly. Hypernyms where distributed like hyponyms and synonyms. Holonyms were distributed like meronyms. Words with the same stem were distributed with

[5]The Google news Word2Vec model lists the semantic distance between *Rohto Pharmaceutical* and *industries* at .494

a rough Gaussian curve.

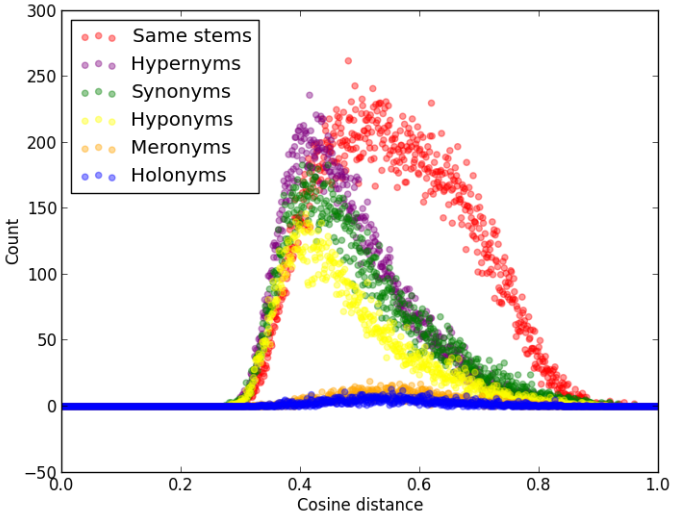

Figure 2: Relations are not distributed equivalently or uniformly across cosine distances. Cosine distances range from 0 to 1.

4.3 Adjusted Frequencies

We sampled 10,000 words in WordNet and found that any given word in WordNet has, on average, more hyponyms and synonyms than meronyms and holonyms. This follows partially from the structure of WordNet. Because WordNet is organized hierarchically as a tree (as shown in Figure 3 from the Computational Linguistics and Psycholinguistics Research Center [17]), any time a node branches, it creates n children. Each of the n children represents n hyponyms (the child nodes). But each node has exactly one parent. Thus, so long as the tree branches, the number of hyponyms for a given non-terminal node will outnumber its holonyms. We cannot account for the relative differences in meronyms, synonyms and holonyms from the structure of WordNet alone.

If some relations have more associated words than others, Word2Vec has more opportunities for finding a hit. This experiment seeks to measure Word2Vec, not WordNet. Thus, we adjust the frequencies from section 4.4 to account for varying numbers of types of relations.

To find an adjusted frequency, we average the averages for each relation to get an overall average. For each relation, we find the ratio of its the average

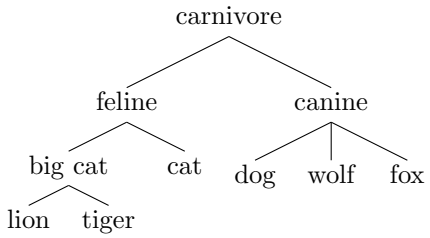

Figure 3: The downward-branching structure of WordNet shows why words have more hyponyms than hypernyms.

Relation	Average Related Synsets in Wordnet
Hyponym	19.7
Synonym	5.0
Hypernyms	4.8
Meronym	3.5
Holonym	1.7
Average	6.94

Figure 4: For a given word, there are different average numbers of related sysets in WordNet.

to the overall average. Then we multiply each (unadjusted) frequency by the inverse of the ratio. We repeat this at all cosine distances. For instance, for any average word, there are roughly 1/3 fewer total holonym words than average. So we multiply the holonym frequency at a given cosine distance by the inverse of this ratio (roughly 3).

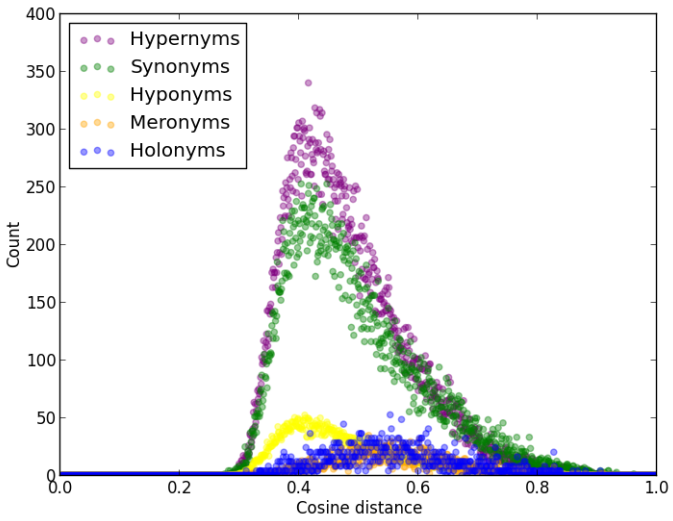

Figure 5: For any given word in WordNet, the average number of words for each relation is dramatically different. Words have more hypernyms and holonyms than holonyms, for instance. We account for this with adjusted counts.

Adjusted counts give a different perspective on Word2Vec. They show that, in general, Word2Vec returns more hypernyms than synonyms – followed by far fewer hyponyms, holonyms and meronyms.

4.4 Cumulative Frequencies

Word2Vec returns words projected into a high-dimensional space. This opens an immediate question: what is the relationship of words that fall within a given semantic distance? In other words, if you were to draw an n-sphere around a given word in Word2Vec's high-dimensional output what kinds of related words would you find. We examine this question by calculating the cumulative frequencies for each relation at different semantic distances.

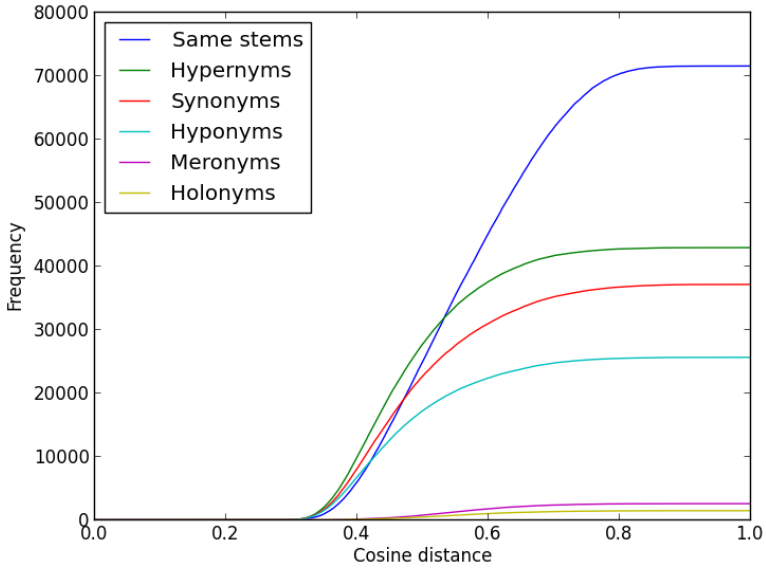

Figure 6: The total frequency of each relation beneath a given threshold levels off across all categories as the semantic distance increases.

4.5 Probabilities

Researchers who try to find particular kinds of relations using Word2Vec need to understand the baseline probability of the the relation at a particular semantic distance. An effective holonym detector, for instance, should beat the average probability of a holonym at a given semantic distance.

Our experiment has determined such probabilities at different cosine values.

We present such probabilities in table 2.

4.6 Jaccard index

In our experiment, we use a binary measure of relatedness: if relevant synsets overlap, we consider the words related. Our experiment would determine that *introduction* and *initiation* are synonyms because their synsets overlap with the synset 'initiation.n.01'. This opens a potential complicating problem: the simple binary determination does not take into account that the word "introduction" has seven associated synsets but the word "initiation" has only 4 synsets – and that these two sets overlap on a single synset. To account for the relative degree of overlap, we also take the Jaccard index of the overlapping sets to gain a better sense of the degree to which semantically related words are related. Our findings are shown in figure 7.

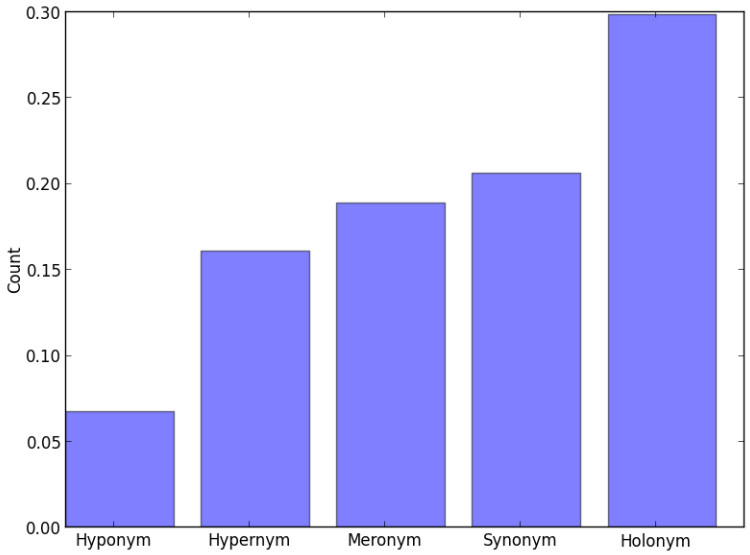

Figure 7: Jaccard indexes vary depending on type of relation – but the variation is closely correlated to the different sizes of synsets in WordNet

At first glance, figure 7 seems to show that Jaccard indexes vary. However, in section 4.3 we sample 10,000 words from the Reuters corpus and find the average number of synonyms, hypoynms, hypernyms, holonyms and meronyms associated with each word (excluding cases where zero relations

are found). Looking over both results, we see that Jaccard indexes for a given type of relation are strongly linked with the average synset sizes for each type of relation in WordNet.

In other words, the Jaccard index for holonyms is the highest among the relations – but the holonymm set has the lowest average size in WordNet. Simiarly, the Jacard index for hyponyms is the lowest – but hyponyms have the most associated words in WordNet. The union of two sets forms the denominator in the calculation of the Jaccard index. Thus, if there are higher numbers of one type of relation we would expect its Jaccard index to be lower (because the denominator would be larger).

Differences in Jaccard indexes seem largely attributable to differences in synset sizes in WordNet (or perhaps in English), not to differences in the semantic relations uncovered by Word2Vec.

5 Future Work

Comparing WordNet and Word2Vec is a limited topic and it has been thoroughly covered here. Most future work on Word2Vec will involve building tools and algorithms using word vectors. This study provides a clear baseline for such efforts. We hope that others use it to benchmark tools.

That said a few matters of comparison have been left uncovered.

This study has only confined itself to the relationship between WordNet and Word2Vec among close neighbors in vector space, the 200 words in Word2Vec. Casting a wider net might yield different reuslts – especially for relations like h

Additionally, this study uses one measure of semantic distance, the cosine distance – which is explained in section 3.However, Word2Vec projects word vectors into semantic space. Thus, the system allows for any number of different measures of geometric distance, like Euclidean distance or Manhattan distance. Such geometric distances might yield different results.

Word2Vec tracks the part of speech associated with a synset. We do not keep track of or use this information in this experiment. However, a follow up experiment that examines how differences in part of speech impact results could yield insights into both Word2Vec and distributional semantics. After all, the fundamental hypothesis underlying distributional semantics is that words that appear together in language – like *piano* and *keys* – have

related meanings. This might be true for nouns or verbs but might not be as true for adjectives or adverbs. The word *good* is used to describe many different things. There is no particular reason to think that it is semantically closer to the word that it modifies. Note that in Word2Vec, words are not assigned a part of speech, so in some cases the contextual part of speech will be ambiguous.

Finally, this study only considered the text of news articles. It would be interesting to compare Word2Vec's performance on different sorts of text. Do news articles tend to favor words at higher levels of generality (hypernyms) over words at lower level of generality (hyponyms)? If different kinds of raw text contain kinds of language, Word2Vec might find different sorts of most-similar words. Thus experimenters might find different results.

6 Conclusion

There are many ways that words may be semantically related. We show that certain semantic relations are more probable than other semantic relations in output from Word2Vec. More precisely, we show that for some word w, we find the probability that a neighbor at a given cosine distance is a synonym, a hypernym, a holonym or a meronym. Our conclusions server as a baseline for further research.

Increment	Synonyms	Meronyms	Holonyms	Hypernyms	Hyponyms	Same stem	None
0.01	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.029	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.049	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.069	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.089	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.109	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.129	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.149	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.169	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.189	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.209	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.229	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.249	0.0	0.0	0.0	0.0	0.0	0.0	1.0
0.269	0.004	0.0	0.0	0.0	0.0	0.0	0.996
0.289	0.013	0.0	0.0	0.011	0.0	0.006	0.961
0.309	0.014	0.0	0.0	0.012	0.0	0.008	0.95
0.329	0.017	0.0	0.0	0.016	0.0	0.009	0.937
0.349	0.017	0.0	0.0	0.015	0.0	0.011	0.935
0.369	0.018	0.0	0.0	0.015	0.0	0.013	0.932
0.389	0.019	0.0	0.0	0.015	0.0	0.016	0.926
0.409	0.02	0.001	0.0	0.015	0.0	0.019	0.92
0.429	0.02	0.001	0.0	0.015	0.0	0.024	0.914
0.449	0.022	0.001	0.001	0.016	0.0	0.028	0.905
0.469	0.023	0.001	0.001	0.015	0.0	0.033	0.9
0.489	0.022	0.002	0.001	0.015	0.0	0.037	0.897
0.51	0.021	0.002	0.001	0.014	0.0	0.042	0.894
0.53	0.021	0.002	0.001	0.013	0.0	0.047	0.89
0.55	0.02	0.003	0.001	0.012	0.0	0.048	0.892
0.57	0.02	0.003	0.002	0.011	0.0	0.052	0.891
0.59	0.019	0.003	0.001	0.011	0.0	0.06	0.883
0.61	0.02	0.003	0.002	0.012	0.0	0.065	0.876
0.63	0.02	0.003	0.001	0.011	0.0	0.076	0.869
0.65	0.021	0.003	0.001	0.012	0.0	0.086	0.856
0.67	0.022	0.003	0.002	0.013	0.0	0.097	0.843
0.69	0.021	0.003	0.001	0.011	0.0	0.101	0.846
0.71	0.019	0.003	0.002	0.01	0.0	0.106	0.847
0.73	0.019	0.003	0.002	0.011	0.0	0.11	0.843
0.75	0.021	0.002	0.001	0.009	0.0	0.121	0.831
0.77	0.024	0.003	0.002	0.01	0.0	0.125	0.821
0.79	0.025	0.002	0.002	0.011	0.0	0.126	0.818
0.81	0.03	0.003	0.002	0.012	0.0	0.127	0.812
0.83	0.034	0.005	0.005	0.015	0.0	0.111	0.811
0.85	0.039	0.001	0.001	0.013	0.0	0.085	0.84
0.87	0.029	0.001	0.001	0.011	0.0	0.059	0.888
0.89	0.02	0.0	0.0	0.007	0.0	0.016	0.947
0.91	0.034	0.004	0.002	0.009	0.0	0.015	0.928
0.93	0.022	0.0	0.0	0.0	0.0	0.036	0.934
0.95	0.0	0.0	0.0	0.0	0.0	0.031	0.969
0.97	0.027	0.0	0.0	0.0	0.0	0.0	0.973
0.99	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Table 2: This table demonstrates the empirical probabilities of finding a particular semantic relation at a particular cosine distance. Researchers may use such a table to establish an empirical baseline when seeking particular semantic relations.

References

[1] word2vec:tool for computing continuous distributed representations of words. https://code.google.com/p/word2vec. Accessed: 2014-11-8.

[2] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[3] Wordnet: A lexial database for english. http://wordnet.princeton.edu/. Accessed: 2014-10-12.

[4] Dominic Widdows. *Geometry and meaning*. Center for the Study of Language and Inf, Stanford, CA, 2004.

[5] Marek Rei and Ted Briscoe. Looking for hyponyms in vector space. *CoNLL-2014*, page 68, 2014.

[6] J R Firth. *Papers in Linguistics*. Oxford University Press, 1957.

[7] Theories of meaning. http://plato.stanford.edu/entries/meaning/. Accessed: 2014-11-8.

[8] Huizhen Wang. Introduction to word2vec and its application to find predominant word senses. 2014.

[9] Bai Xue, Chen Fu, and Zhan Shaobin. A study on sentiment computing and classification of sina weibo with word2vec. In *Big Data (BigData Congress), 2014 IEEE International Congress on*, pages 358–363. IEEE, 2014.

[10] Mohit Iyyer, Peter Enns, Jordan Boyd-Graber, and Philip Resnik. Political ideology detection using recursive neural networks. In *Association for Computational Linguistics*.

[11] Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[12] Christiane Fellbaum. Wordnet and wordnets. In Keith et al. Brown, editor, *Encyclopedia of Language and Linguistics, Second Edition*. Elsevier, Oxford, 2005.

[13] D Alan Cruse. *Lexical semantics*. Cambridge University Press, 1986.

[14] Tony Rose, Mark Stevenson, and Miles Whitehead. The reuters corpus volume 1-from yesterday's news to tomorrow's language resources. In *LREC*, volume 2, pages 827–832, 2002.

[15] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.

[16] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA.

[17] Computational linguistics and psycho linguistics research center. http://www.clips.ua.ac.be/pages/pattern-search. Accessed: 2014-11-10.

7 Vita

Abe Handler currently works as a software developer and data journalist for *The Lens*, an online, non-profit investigative newsroom. He entered the Masters program in Computer Science at the University Of New Orleans in 2012. He is interested in information retrieval, natural language processing, machine learning and data science – especially as it pertains to law, politics, social science and journalism. Abe holds a B.A. in philosophy from Columbia University, where he reported for the student newspaper *The Columbia Daily Spectator* – graduating with honors in 2007.