Graphical Analysis of the Wordnet Lexicon

Atticus Geiger
atticusg@stanford.edu

Sandhini Agarwal
sandhini@stanford.edu

Abstract

We investigate the structure of nouns in the English lexicon using the Word Net data base. We construct two classes of graphs, one class with meanings as nodes and the other with words as nodes. For both classes, we construct edges based on the lexical semantic relations of hypernymy, polysemy, and meronymy. We characterize the global structure of these graphs, finding a small world structure emerges when the polysemy and hypernymy relations are considered together. We also conduct a mesostructural analysis including structural role discovery using the RolX algorithm, community detection using the Louvain algorithm, and node embedding construction using Poincare and node2vec algorithms. We conduct an analysis to determine whether there is interactions between the polysemy and hypernymy relation, and discover some evidence that there is. We additionally test the viability of our node vectors for the task of natural language inference, and find weak evidence that using such vectors can increase the generalization capabilities of neural models.[1]

1 Introduction

Polysemy is the crosslinguistic phenomenon of individual words being mapped to multiple distinct meanings. Polysemy often connects meanings that do not have an interesting semantic relation, for example institutions where we store our money and the land next to a river are concepts with no profound semantic connection, but the word *bank* has both meanings. As such, it is not obvious whether polysemy occurs arbitrarily or has some deep causes governing it. To investigate this question, we analyze the role of polysemy in the structure of the English lexicon and whether it is influenced by the relations of hypernymy and meronymy.

The English Lexicon consists of all English word meanings and the relations between them. The part of speech we consider here is nouns and the relations we consider are hyponymy/hypernymy, meronymy/holonymy, and polysemy. A hypernym is a word meaning that is broader than its hyponym. For example, *animal* is a hypernym of *dog*. A meronym is a meaning that is a part of its holonym's meaning. For example, *finger* is a meronym of *hand*. We consider two meanings to be in the polysemy relation if there is a polysemous word that has both meanings. We also define these relations over words. We define two words to be in the hyponymy relation if any of their meanings are in the hyponymy relation, and likewise for meronymy. We define two words to be in the polysemy relation if they share a meaning in common.

We aim to study polysemy using the following methods. First, we will construct different graph types to capture different relations between words. We will then carry out analyses of these relations using role discovery and community detection. We will use this to assess the relationship between hyponymy, meronymy and polysemy. Then, we will study if we can predict whether a word is a polysemy using node embeddings trained using the hyponymy graph. These methods will help us establish either the presence or the lack of a correlation between hyponymy and polysemy.

We additionally test the viability of node vectors trained on the hypernymy task for the task of natural language inference. We hope to assess if capturing the structural relations within language itself can impact progress in NLP and provide evidence for it.

2 Related Work

Sigman and Cecchi (2002) have investigated the

[1] Github Repo- https://github.com/atticusg/WordNetProject

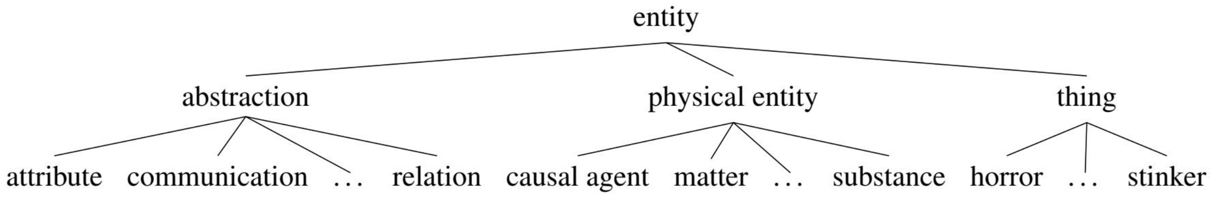

Figure 1: The tree induced on noun meanings by the hypernymy relation

global structure of the Wordnet lexicon. They found that the three semantic relations of hypernymy, meronymy, and polysemy are scale invariant, which is typical of naturally occurring self organizing graphs. They began with the hypernymy relation, which creates a tree structure over the set of nouns and a large average minimal path. They found that the inclusion of the polysemy relation transformed the graph into a small world network (Watts and H. Strogatz, 1998). Moreover, they found that that the length of minimal paths between nodes in the hypernymy tree structure show low correlation with the length of minimal paths between the same nodes once polysemy is added. They also identified the three largest simplexes, resulting from the highly polysemous words *head*, *line*, and *point*, in as the traffic hubs of the network.

We see an opportunity to add breadth and depth to the work of Sigman and Cecchi (2002). Wordnet is a growing database and we reproduce results on scale invariance, minimal paths, and clustering. We discover subgraph communities and identify various structural roles nodes play. Finally, we consider a new class of graphs where words are nodes and provide the same global and mesostructural analysis on these graphs.

3 Dataset

For our analysis, we use the database Wordnet, an impressive representation of the English lexicon (Fellbaum, 1998). In Wordnet, a meaning is represented as the set of words that have that meaning. Such sets are called synsets. For example, the meaning of a long seat with arms with room for two or more people is represented as the synset $\{couch, sofa, lounge\}$. The word $couch$ is also contained in the synset for the meaning of phrasing of expressing something in a specific manner, e.g. "His comments were couched in strong terms". Then we would consider these two meanings to be in a polysemy relation, as the word $couch$ can evoke both of them. This example shows how Wordnet encodes the polysemy relation.

Graph	Nodes	Edges	P	C
G_M^H	82115	84427	13.06	0.00048
G_M^M	82115	22187	14.24	0.0015
G_M^P	82115	60662	9.30	0.21
G_M^{HP}	82115	145064	7.66	0.11
G_M^{HMP}	82115	166483	7.26	0.11
G_W^H	117798	300890	8.52	0.014
G_W^M	117798	101021	10.86	0.017
G_W^P	117798	108771	9.77	0.37
G_W^{HP}	117798	408512	6.34	0.74
G_W^{HMP}	117798	506651	2.42	0.74

Table 1: Statistics characterizing the global structure of our graphs where P is average minimal path and C is the average clustering coefficient.

Wordnet also contains hypernymy and meronymy relations between meanings. The hypernymy relation defines a tree-like structure over the set of all noun meanings. The meaning of the word *entity* according to wordnet is 'that which is perceived or known or inferred to have its own distinct existence (living or nonliving)' and it is this meaning that is the root of the tree. In figure 1, we show the first few levels of this tree. The meronymy relation is divided into three subcategories, but we ignore these for our analysis.

At this point in time, Wordnet contains 82115 meanings and 117798 words, but these numbers are arbitrary and ever growing.

4 Graph Construction

We find two natural sets of nodes in Wordnet the set of all meanings and the set of all words. For each of these sets of nodes, we have three sets of edges corresponding to the hypernymy, meronymy, and polysemy relations. In this paper, we denote graphs by the symbol G with subscripts and superscripts. If the subscript W is present, the nodes of the graph are words and if the subscript M is present the nodes of the graph are meanings. Similarly, if the superscripts H, M, and/or P are present, the edges of the graphs are from the relations hyponymy, meronymy, and polysemy, respectively. For example, G_W^{HP} is a graph where words are nodes and edges are defined by the hypernymy and polysemy relations. All graphs we consider are undirected. We will treat these graphs as simple graphs, except in structural role discovery where we will treat G_W^{HMP} and G_M^{HMP} as multigraphs and for training poincare embeddings, we will treat G_M^H as directed.

5 Global Organization

In this section, we characterize our graphs using global properties. We begin with basic terminology. The degree of a node is the number of edges the node has and we will sometimes use hypernymy/meronymy/polysemy degree to refer to the number of edges a node has from a particular relation. The density of a graph is the fraction of edges that exist out of all possible edges and is computed as $\frac{2|E|}{|V|(|V|-1)}$.The average path length is the average of all minimal paths in a graph, computed as $\frac{1}{|V|(|V|-1)}\Sigma_{i,j\in V} dist_{min}(i,j)\cdot\delta_{i,j}$ where δ is the Kronecker delta. The clustering coefficient of a node is the fraction of edges between the neighbors of a node out of all possible edges, and is computed as $\frac{2e_i}{k_i(k_i-1)}$ for a node i with degree k_i and e_i edges between the neighbors of i.

In table 1 we find the nodes, edges, density, average minimal path length, and average clustering coefficient of 10 graphs. We observe the graph G_M^{HMP} has a small average minimal path, high clustering coefficient, and low density which means it is a small world network, reaffirming the conclusion of Sigman and Cecchi (2002) on this current interation of Wordnet (Watts and H. Strogatz, 1998). We also observe that G_W^{HMP} is a small world network, with an even larger clustering coefficient and lower average minimal path. We deduce that word nodes are more clustered and have a lower diameter because words adopt all the relations of their multiple meanings, resulting in the number of total edges being significantly larger in the graphs with word nodes.

In Figure 2 we show the hyponymy, meronymy, and polysemy relations between words and between meanings are scale invariant. This is typical of self-organizing naturally occuring networks.

We investigate the relationship between hypernymy and polysemy in Figure 3, which plots pairs of meanings in the polysemy relation against the minimal path between the meanings in the hypernymy tree in Figure 1. We can see that the distribution of Wordnet data is to the left of the distribution of the graph with randomly generated polysemy relations. This indicates that if two meanings are closer in the hypernymy tree, then they are more likely to be in the polysemy relation. This is the first piece of evidence we discovered supporting the idea that hypernymy influences polysemy.

6 Methods

6.1 Community Detection

A community in a graph is a set of highly connected nodes and the community detection algorithm we use is the Louvain Algorithm which attempts to maximize the modularity of communities. This algorithm considers communities in a graph to be nodes with a high modularity, which is a quantification of how many more edges occur in a set of nodes than one would expect. The modularity of a graph G with partition P is quantified as follows:

$$Q(G,P) = \frac{1}{2m}\Sigma_{p\in P}\Sigma_{i\in p}\Sigma_{j\in p}(A_{ij} - \frac{k_i k_j}{2m})$$

Where A_{ij} is the weight of the edge between i and j, k_i and k_j are the degrees of i and j, and $2m$ is the sum of all the edge weights in the graph.

We now describe the Louvain Algorithm, which greedily maximizes modularity with local changes in community membership (Blondel et al., 2008). To begin, nodes are all put in their own separate communities. Then, we repeat the following two phases until there is no further increase in modularity. The first phase loops through every node in random order and computes the changes in modularity that would result from putting that node in any other community. The node is then put into the community that results in the largest positive change in modularity. This process is repeated until there is no movement that would yield a gain in modularity. The second phase contracts the partitions from the first phase into super nodes, where two super nodes are connected if their corresponding partitions contain nodes that are connected. The weight of an edge between two super nodes is the sum of the weights from all edges between

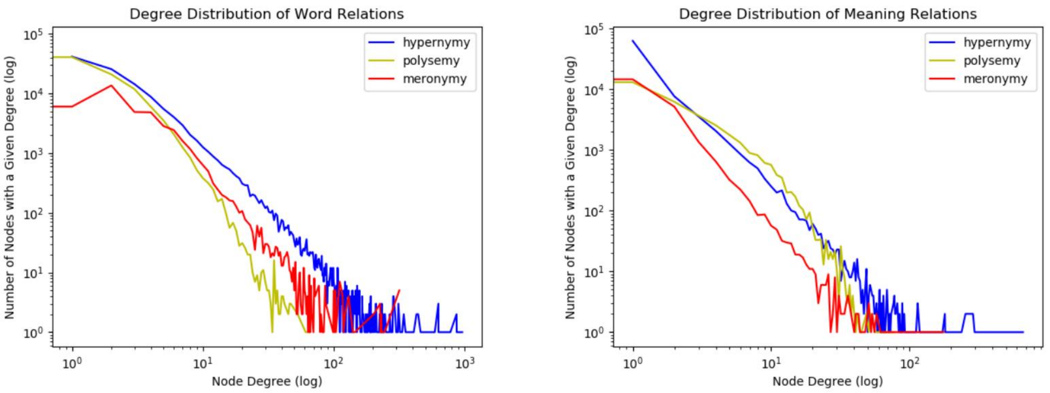

Figure 2: The scale invariant distribution of relations between meanings and between words. The log-log plot shows linear dependence between number of nodes and degrees, demonstrating power law behavior. The relations between words have a higher degree

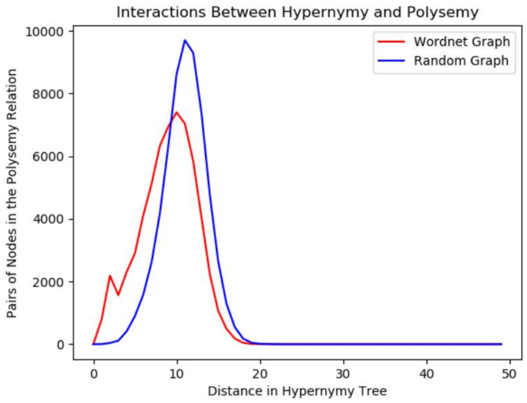

Figure 3: The distribution minimal paths in the hypernymy tree of Figure 1 between pairs of meanings in the polysemy relation. The data from Wordnet is compared against a randomly generated polysemy graph with the same number of edges.

their corresponding partitions. The output of the second phase is this super node network.

The Louvain algorithm provides a hierarchy of partitions. The hypernymy relation naturally sorts meanings into a tree hierarchy, as described in section 3 and seen in Figure 1. We run the Louvain algorithm on the graph G_M^P to attain a different hierarchy of meanings from the polysemy relation. A priori, we do not know whether these hierarchies will be at all similar so in our analysis, we compare the two.

6.2 Structural Role Discovery

We use a the RolX algorithm adapted to multigraphs to compute feature vectors and perform role discovery (Henderson et al., 2012). We run this algorithm on G_M^{HMP}, which we treat as multigraphs. At a high level, the RolX algorithm recursively creates node feature vectors that encode information about the structure of the graph around the node.

We now present the RolX algorithm. We begin with 9 dimensional basic feature vectors for every node consisting of the nodes hypernymy degree, meronymy degree, and polysemy degree, the number of hypernymy edges, the number of meronymy edges, and the number of polysemy edges in the nodes egonet, and the number of hypernymy edges, the number of meronymy edges, and the number of polysemy edges connecting from the node's egonet to the rest of the graph.

Then these feature vectors are recursively expanded. Each recursive step takes the current feature vector of a node and appends the summation and the mean of the feature vectors of the node's hypernymy, meronymy, and polysemy neighbors. This process grows the dimensional of feature vectors exponentially, so at each recursive step we remove features with a correlation score greater than 0.9. Once this process creates rich feature vectors for nodes, we use non-negative matrix factorization to group nodes into structural role groups. We limit the number of recursions based on our computational resources. To determine the number of roles, we increased the number of roles until there were two roles that did not have obvious differences from one another. We arrived at 8 roles.

6.3 Node Embedding Analysis

A node embedding is a distributed representation of a nodes structural role in a graph. We carry out two series of experiments using Wordnet for generating node embeddings. The first experiment aims to study if there is a structural relation between graphs constructed using hypernymy relations and those constructed using polysemy relations. The second experiment aims to study if adding information from hypernymy relations has the potential to improve the performance of existing embeddings on certain tasks such as natural language inference.

6.3.1 Experiment 1: Poincare Embeddings

In this experiment, we trained embeddings from the G_M^H graph using the poincare technique. Poincare embeddings are well suited for making use of structural and heirarchical linkages in graphs.

Poincare embeddings compute embeddings in hyperbolic space as opposed to in Euclidean space. Hyperbolic space has a constant negative curvature and this can informally be equated to a tree structure and as a result is well suited for hierarchical structures.(Nickel and Kiela, 2017)

On a high level, poincare embeddings capture hierarchical structures because they account for two notions of similarity. Firstly, they aim to place nodes that are similar to one another close to each and nodes that are dissimilar far from each other. Secondly, they also account for hierarchy by trying to place nodes lower in the hierarchy further away from the origin and nodes that are high close to the origin.(Nickel and Kiela, 2017) Thus, in our case when we train embeddings for hypernymy relations, parent nodes or root nodes such as 'entity' should be close to the origin and their children, nodes such as 'causal agent', should be nearer the edges.

The hyperbolic distance between two points is given by -

$$d(u,v) = arcosh(1 + 2(\frac{||u-v||^2}{(1-||u||^2)(1-||v||^2)}))$$

We use these embeddings trained on the G_M^H graph to then carry out link prediction and graph reconstruction on the G_M^P graph. We chose poincare embeddings for this task as we are trying to assess if the hierarchical nature of hypernymy in particular has an impact on polysemy relations. The performance of embeddings trained solely on G_M^H graph on link prediction and graph reconstruction on G_M^P graph has the potential to give us clues as to whether there can be a structural relation between polysemy meanings and hypernymy relations. Again, a priori we have no indication of what results to expect since linguists and psychologists have not yet made conclusive claims as to how polysemy and hypernymy may be related.

6.3.2 Experiment 2: Node2Vec Embeddings

We use the algorithm node2vec to create node vectors using the graph G_W^H. At a high level, node2vec optimizes the vector representation of a node n to have a high dot product with the vectors of nodes that are passed through during random walks starting at the node n. The algorithm DeepWalk uses completely randomized walks, and node2vec uses walks generated with two parameters p and q. During a random walk, the unnormalized probability of transitioning to a node is $\frac{1}{p}$ if that node is closer to the origin, 1 if that node is equidistant from the origin, and $\frac{1}{q}$ if that node is further from the origin. When a random walk is run, we collect the multiset of nodes reached. It then optimizes the embeddings using stochastic gradient descent. The algorithm for loss function is as follows:

$$\mathcal{L} = \Sigma_{u \in V} \Sigma_{v \in NR(u)} (-log(P(V \mid ZU))$$

When a graph has words for nodes, node vectors can be used as word vectors in NLP tasks. There exists a large literature on the creation of word vectors, with the prominent word vectors being GloVe and word2vec (Pennington et al., 2014; Mikolov et al., 2013). Tasks such as natural language inference (NLI) rely greatly on the ability to recognize lexical relations such as hypernymy, so there is potential for these word net node vectors to be useful in natural language understanding tasks.

We use the embeddings we trained on the G_W^H graph and append them to existing GloVe vectors to study if they impact the performance of models on NLI tasks.

We chose to use node2vec embeddings for this task, because as a first step, we wanted to study if the hypernymy linkages alone- without the additional information about the hierarchy in which they're organized- would be sufficient for an increase in the performance on tasks such as NLI. As a next step, other embedding techniques such as poincare embeddings can also be tested for performance.

7 Results

7.1 Structural Role Discovery

Using the RolX algorithm, we identified 8 structural roles for meanings treating the graph G_M^{HPM} as a multigraph. We manually inspected 20 randomly chosen nodes from each role to characterize it. Role 1 contains 140 meanings that are in or closely connected to large cliques in the graph G_M^P, includings the various meanings of *head*, *line*, and *point* that Sigman and Cecchi (2002) identified as the traffic hubs of the network. For example, other meanings in Role 1 are the various meanings of *mind* and brain which have polysemous links to the meanings of *head*. Role 2 consists of 12 nodes that are part of highly connected supgraphs in G_M^M. Role 3 contains 103 meanings with very high degrees in the graph G_M^H. Role 4 contains 1370 meanings with high betweeness in the graph G_M^H. Role 5 contains 9726 meanings disconnected from the main hypernymy tree. Role 6 contains the 12039 meanings in the strongly connected component of the graph G_M^P. Role 7 contains 8331 nodes with high meronymy and hypernymy degrees. Role 8 contains all other 50394 nodes.

Unfortunately, our multigraph RolX algorithm did not capture any interactions between the relations hypernymy, meronymy, and polysemy except for role 7, which characterized nodes based on both hypernymy and meronymy. This could be because there are not other meaningful ways to characterize a node across multiple relations, or perhaps there are a different extension of RolX to multigraphs is necessary to discover them.

7.2 Community Detection

We used the Louvain algorithm on the graph G_M^P to create a hierarchy of meanings that we can compare to the hierarchy of meanings created by the hypernymy relation. We chose to do this analysis on the graphs with meanings as nodes, because in the graphs with words as nodes, the hypernymy relation creates a much messier hierarchy of meanings because every word has hypernyms and hyponyms for each meaning it can have.

We chose the following way to compare the hypernymy hierarchy and the polysemy hierarchy created by the Louvain algorithm. We consider the lowest common hypernym of a given community, which is the meaning in the hypernymy tree that is furthest from the root node and is a hypernym of every meaning in the community. Once we have the lowest common hypernym of a community, we compute its minimum distance from the root node of the hypernymy tree for meanings. The larger this minimum distance is, the closer the nodes of the community are in the hypernymy tree.

Iteration	G_M^P	Configuration Graph
1	1.61	0.60
2	1.75	0.10
3	2.21	0.20
4	2.36	0.40
5	2.38	0.94
6	-	1.09

Table 2: The average minimum distance of the lowest common hypernym across all communities in a given iteration of the Louvain algorithm. Results are provided for the graph G_M^P and a configuration graph made based on G_M^P.

In Table 2, for a given iteration of the Louvain algorithm we provide the average minimum distance of the lowest common hypernym across all communities. We additionally provide a configuration graph as a control. The communities formed in the first iteration are the cliques that single polysemous words create, e.g. the word *head* has 31 meanings and all of those meanings are connected to one another forming a polysemy clique of size 31. The further iterations of the algorithm finds communities by grouping this cliques together.

We can see across all iterations the polysemy graph has a higher average minimum distance than the configuration graph, and so we can conclude that the community structure of polysemy is linked to the structure of the hypernymy graph. What is more notable is the fact that the average minimum distance increases across iterations in the polysemy graph and the largest increase in average minimum distance is between the second and third iterations. This tells us that cliques resulting from polysemous words are less in accordance with the hypernymy tree than the larger community structure connecting those cliques. This evidences that the information of hypernymy graph structure will be worse for predicting individual polysemy relations than larger structural groupings.

Model	Train	Test	Adversarial Test
LSTM encoder w/ Glove	78.4	71.2	19.1
LSTM attention w/ Glove	79.2	73.3	22.3
LSTM encoder w/ Glove + Node Vectors	77.9	71.1	25.1
LSTM attention w/ Glove + Node Vectors	79.2	73.6	26.3

Table 3: The accuracy of an LSTM encoder model and LSTM attention model on the SNLI dataset using only GloVe word vectors and using both GloVe word vectors and node vectors from the graph G_M^H. We additionally test on an adversarial test set that requires learning simple lexical relations between words.

7.3 Node Embeddings

7.3.1 Poincare Embeddings

We created node embeddings using the poincare embeddings technique on the G_M^H graph, the G_M^P graph and a graph with edges between random nodes. Unlike the G_M^H graph, the G_M^P graph and the random graph do not have a hierarchical structure. However, we still computed these embeddings in order to compare our results for link prediction and graph reconstruction on G_M^P.

We tested to see how link prediction on the G_M^P graph would work when trained using embeddings trained from the G_M^H graph. Our train set comprised of 14440 polysemy edges and our test set of 182 polysemy edges.

We got a mean average precision (MAP) of 0.7113 on G_M^P when we carried out link prediction using embeddings from G_M^H. This precision is significantly lower than the precision we received when carrying out link prediction on G_M^P using embeddings trained on G_M^P. However, it is difficult to conclude from this evidence alone if there is a relation between G_M^P and G_M^H.

We also created a graph with links between random nodes. We then test to see how embeddings trained on this random graph performed on the link prediction task for G_M^P. The aim was to study if there is a noticeable difference in the accuracy between embeddings trained on the random graph and the G_M^H graph. Surprisingly, we found that G_M^H performs significantly worse than embeddings trained on the random graph. The random graph gives a MAP of 0.86. Thus, this is evidence that the structural nature of hypernymy graphs and polysemy graphs do seem to have some relation. While our experiments alone don't reveal what the nature of this relation might be, we do know that the relation between the two is not random.

We also tested to see how graph reconstruction would perform on the G_M^P graph when trained on embeddings from the G_M^H graph. We got a mean average precision of 0.852 on the G_M^P graph when using embeddings trained on G_M^H and a precision of 0.95 when using embeddings trained on G_M^P. Again, this alone is insufficient to draw a conclusive relation between G_M^H and G_M^P.

Embedding	G_M^P MAP
G_M^P	0.88397
G_M^H	0.71138
Random Graph	0.86116
	G_M^P Mean Rank
G_M^P	1.439
G_M^H	2107.307
Random Graph	839.259

Table 4: Graph showing link prediction results on G_M^P

Embedding	G_M^P MAP
G_M^P	0.95008
G_M^H	0.85211
Random Graph	0.92690
	G_M^P Mean Rank
G_M^P	1.714
G_M^H	2486.322
Random Graph	820.934

Table 5: Graph showing graph reconstruction results on G_M^P

When we used embeddings trained on random graph for reconstruction, we got a precision of 0.92. This is also higher than the precision of G_M^H which was 0.85. This indicates that there is some sort of relation between hypernymy and polysemy as the perfomance is not random. Hypernymy relations are not good predictors for polysemy relations and are, in fact, worse than random. While we don't have a theory explaining this behavior, this demonstrates that there is some nature of interaction between the two.

Our results from the runs are demonstrated in Table 4 and Table 5.

7.3.2 Node2Vec Embeddings

We also create node embeddings using the node2vec algorithm. We ran 100 walks per node with a walk size of 20 and a window size of 20. We used a p value of 1000000 and a q value of 1 so random walks will reach distant hypernyms and hyponyms. We ran the node2vec algorithm with these parameters on the undirected graph G_W^H and two directed versions of G_W^H, one where hypernyms point to hyponyms and the other where hyponyms point to hypernyms. For each word, we got three 50 dimensional node vectors. We used two directed versions of the graph to capture the asymmetry of the hypernymy relation; using the undirected graph alone, node vectors for *dog* and *animal* would have a high dot product, but there would be no information informing which was the hypernym and which was the hyponym.

We chose the task of natural language inference (NLI) to test the usefulness of these word vectors. The three class conception of NLI involves categorizing a premise and hypothesis sentence into three categories, *entailment* if the premise being true means the hypothesis is true, *contradiction* if the premise being true means the hypothesis is false, and *neutral* otherwise. To perform the task of NLI, it is often necessary to recognize hypernymy and hyponymy relations between words in the premise and hypothesis. The dataset we use is the Stanford Natural Language Inference corpus (SNLI), a recently created large scale NLI dataset on which neural models are state-of-the-art (Bowman et al., 2015). The models we consider are the LSTM encoder model of Bowman et al. (2015) and the attention LSTM model of Rocktäschel et al. (2015), which is designed to identify lexical relationships for use in inference. We additionally test on an adversarial test set provided by (Glockner et al., 2018), which is specifically designed to test the abilities of models to generalize to examples requiring new lexical relations that were not seen in training.

We provide the results of our NLI experiments in Table 3. We test both models only using GloVe word vectors and using GloVe word vectors concatenated with the three 50 dimensional node vectors we created. For words that we do not have node vectors for, such as adjectives or verbs, we append a 150 dimensional random vector. We can notice that the inclusion of our node vectors does not seem to have an impact on the normal SNLI test set, but does result in an increased performance on the adversarial test set. This evidences that inluding our node vectors increase the generalization capabilities of neural NLI models.

We performed only a small hyperparameter search due to our limited computational resources, so these results are far from definite, but we can still look to them for an indication of how viable these vectors could be.

8 Conclusion

Here we investigated the global and mesostructural organization of nouns in the English lexicon with the relations of hypernymy, meronymy, and polysemy. We aimed to use graph theory to shed light on questions surrounding how these relate to one another, and the role they play in comprehension, which linguists and psychologists have attempted to answer for decades.

We constructed two classes of graphs, one class where words are nodes and one class where meanings are nodes. We found that the graphs that include at least the hypernymy and polysemy relations are small world networks, and that the small world networks where words are nodes have a much smaller diameter and higher clustering coefficient than the small world networks where meanings are nodes. Our role discovery did not find significant interactions across the three relations. However, we found that the hierarchy of communities created by the polysemy relation group themselves in accordance with the hypernymy tree, particularly when considering the larger partitions of the hierarchy.

Additionally, we found that node embeddings trained on hypernymy graphs actually perform worse than random when doing link prediction and graph reconstruction on polysmy graphs. While we do not have a theory or evidence for why this happens, this is evidence that the relation between hypernymy and polysemy is not random and that there might be some nature of relation between the two.

Lastly, we found evidence that node vectors trained on a hypernymy graph can result to increased generalization capabilities for neural NLI models, however this result is tentative as we lacked the computational resources to thoroughly investigate the potential of this approach. While these results are tentative, they demonstrate how harnessing the structural relations within language itself has the power to greatly impact progress within NLP.

References

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Christiane Fellbaum, editor. 1998. *WordNet: an electronic lexical database*. MIT Press.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking nli systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655. Association for Computational Linguistics.

Keith Henderson, Brian Gallagher, Tina Eliassi-Rad, Hanghang Tong, Sugato Basu, Leman Akoglu, Danai Koutra, Christos Faloutsos, and Lei Li. 2012. Rolx: Structural role extraction mining in large graphs. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1231–1239.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, pages 6338–6347.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *In EMNLP*.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *CoRR*, abs/1509.06664.

Mariano Sigman and Guillermo A. Cecchi. 2002. Global organization of the wordnet lexicon. *Proceedings of the National Academy of Sciences*, 99(3):1742–1747.

Duncan Watts and Steven H. Strogatz. 1998. Collective dynamics of small world networks. *Nature*, 393:440–2.