Semi-supervised Learning on Small Worlds

Rosie Jones*
Yahoo! Research Labs
74 N Pasadena Ave, 3rd Floor
Pasadena, CA 91103
jonesr@yahoo-inc.com

ABSTRACT

Algorithms for semi-supervised and weakly supervised learning rely on the cooccurrence of features to propagate labels. Multiple-view algorithms, such as cotraining and coEM, use connectivity from the bipartite graph over two complementary feature sets to bootstrap models from a few initial labeled examples. Single-view algorithms, such as EM and relevance feedback (self-training) also rely on connectivity on the graph, but look at all features simultaneously. We show how understanding algorithms, and the data sets they operate on, in graph theoretic terms can explain some of the performance of these algorithms. We give empirical results over an application on learning semantic classes, as well as showing that Blum and Mitchell's data also displays small-world properties. We also show that this enables us to predict the effectiveness of an algorithm on a task, based on the properties of the initial examples, as well as overall connectivity of the feature-cooccurrence graph. This may also allow us to predict how well these algorithms will perform on new datasets with small-world properties, as well as providing a way of recommending initial examples for labeling, and informing active learning algorithms.

1. INTRODUCTION

Semi-supervised and weakly supervised learning algorithms use a combination of labeled and unlabeled data to learn a function. Typically the algorithm is initialized with examples of the target function, which allow a weak model to be built, which can then by used to classify unlabeled examples, which in turn contribute to improving the accuracy of the model. Examples of algorithms displaying these properties include EM, cotraining [1], coEM [8], and metabootstrapping [10].

The standard proof of the effectiveness of cotraining uses graph theoretic arguments over an idealized model on a bipartite graph [1]. Blum and Mitchell perform their analysis under assumptions of statistical independence of feature sets given the target class, and distribution of features according to a random graph model. In particular, their reasoning applies to connected components of a graph, in which each component contains instances from a single class, and the initial labeled set covers all components with high probability.

Nigam and Ghani [8] show that coEM is more effective for semi-supervised learning than both cotraining and EM, both when statistical independence of the feature sets given the target class is satisfied, and when it is not satisfied. Muslea et al. [5] show that cotraining style algorithms can be effective across a range of levels of statistical independence of feature sets.

One of the reasons semi-supervised algorithms are effective is that a model which has accurate estimates from only a subset of the features can label new examples for which some of the features are unknown. It has been argued that the ideal setting for a semi-supervised algorithm is one in which the features are correlated, that is, there is some redundancy in the feature set, but they are statistically independent given the target class, that is, within the target class, a feature is as likely to cooccur with any feature as any other feature [5, 1]. We will show these arguments are not entirely realistic, given data exhibiting small-world properties. We will show that semi-supervised algorithms nevertheless can perform reasonably well on this type of data, and argue that taking into account the small-world structure when labeling initial examples, as well as in active learning, can improve performance.

It has been shown that word cooccurrence graphs and synonymy relationships [2, 11] do not exhibit random graph structure, but rather a small-world structure, with most nodes reachable from most other nodes within two to three steps. We will show that Blum and Mitchell's dataset also exhibits these properties. In addition, we will perform a detailed analysis of graph theoretic properties of a a data set for information extraction, and show strong correlations between algorithm performance and some of these properties.

2. SEMANTIC CLASS LABELING TASK

The data we use for analyzing the effect of graph structure on semi-supervised learning performance comes from the information extraction task of learning to identify locations,

*This research was carried out while the author was at Carnegie Mellon University.

Class	Seeds
locations	australia, canada, china, england, france, germany, japan, mexico, switzerland, united states
organizations	inc., praxair, company, companies, dataram, halter marine group, xerox, arco, rayonier timberlands, puretec
people	customers, subscriber, people, users, shareholders, individuals, clients, leader, director, customer

Table 1: Seeds used for weak labeling of data, for initialization of bootstrapping.

organizations and people in context. This data has been described elsewhere [3, 4]. Each example consists of two features: a noun-phrase n and a local syntactic context c; this data consists of two partially redundant feature sets.

The local syntactic contexts represent both information about the grammatical role the noun-phrase plays in the sentence, and the meaning of that sentence. For example, two distinct contexts are given by "$< x >$ provides" and "$< x >$ provide". The $< x >$ shows where the noun-phrase falls in word-order, relative to the rest of the context. Distinguishing between these cases is useful for discriminative power, since they enable us to distinguish subjects which can only be third-person singular from those which can be plural and those which can be first- or second-person. By contrast "`provide` $< x >$" covers cases in which the target class can be the *object* of the verb provide, rather than the subject. Subjects of the verb `provide` tend to be agents, such as people and organizations, whereas objects of the verb `provide` tend to be resources.

Initial examples are labeled by identifying any noun-phrase with one of the seeds in Table 1 as its head, as positive. We also conduct experiments looking at different candidate seeds for the same task. We took a list of 253 country names (`allcountries`) from a list of country domain names, and took subsets of size 10 and 20 from this list. The number of occurrences of the words in the list was quite variable, as shown in Table 2.

3. SMALL-WORLD GRAPHS OF DATA

We are accustomed to thinking of examples in machine learning as vectors of features, where an example x_i is made up on n features: $x_i =< x_{i_1}..x_{i_n} >$, and may also be accompanied by a label y_i. In this section we describe how we can view a set of examples $X = \{x_1..x_m\}$ as a graph.

We describe both representations of examples with split feature sets as bipartite graphs, as well as more general representation of examples as nodes and edges in a graph. For the remainder of this paper we will concern ourselves with bipartite graphs made up of split feature sets.

We will consider a unique instantiation of a feature or features to be a node in the graph, and cooccurrence among features or feature sets to be an edge in the graph.

Seed Set(s)	Num Seeds	n	σ
10-random	10	32.9∗	40.0∗
20-random	20	74.9∗	59.3∗
orig-10	10	894	
allcountries	253	1016	

Table 2: For the `locations` task, 10 random sets of 10 and 20 country names matched variable numbers of instances in the corporate web-page data. Shown here is the average number of instances matching, across the 10 sets, and the exact number of instances matching for the original 10 country names, and the entire list of 253 country names. The 10 country names used in basic experiments were very frequently occurring. Using all 253 country names from a list of country names did not match many more initial examples. n is the number of examples matching the seeds (marked with $*$ when this is an average calculated over 10 sets). σ is the variance in the number of seeds matching, when the average was calculated over 10 sets.

f_1	f_2	label
australia	flew to $< x >$	?
australia	headquartered in $< x >$	+
australia	$< x >$ broadened	?
china	flew to $< x >$	?
france	headquartered in $< x >$	?
thailand	$< x >$ broadened	+
thailand	gulf of $< x >$	?
director	$< x >$ of multinational company	?
leader	$< x >$ in its industry	?

Table 3: Training examples in feature vector format. Each example has two features, f_1 (the noun phrase) and f_2 (its context). Some examples are labeled positive, while all other examples are unlabelled.

3.1 Bipartite Graphs over Examples Represented with Two Feature Sets

Table 3 shows training examples for a semantic labeling task which we described in Section 2. Each example has two features, f_1 (the noun phrase) and f_2 (the context). Some examples are labeled positive, while all other examples are unlabelled.

In Figure 1 we see a bipartite graph representing the same instances. Each instance is represented by an edge joining two nodes in the graph. For example, the instance "flew to china" is represented by the two nodes "flew to $< x >$" and "china", with an edge joining them. In this graph, the node "australia" is connected to three other nodes, so it has degree 3. This also represents the fact that "australia" occurred in 3 unique examples, with three unique different contexts: `flew to` $< x >$, `headquartered in` $< x >$, and $< x >$ `broadened`.

In supervised machine learning, viewed from this graphical perspective, we are given a set of nodes and edges, with a label provided for each of the edges in a training set. We use these labels to learn to predict labels on edges in a held-out test set graph.

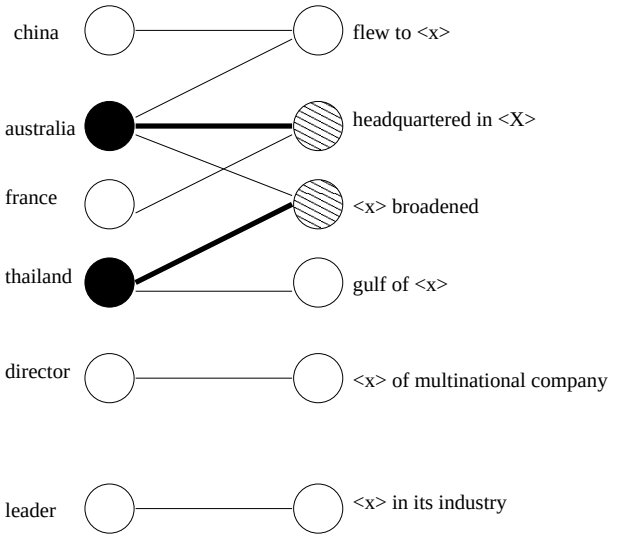

Figure 1: Each instance represents an edge joining two nodes in the graph. For example, the instance "flew to China" is represented by the two nodes "flew to $< x >$" and "china", with an edge joining them.

For this work we are concerned with semi-supervised learning. Our training set is a graph with all edges present. We are given a small number of labeled examples, that is, a small number of labeled edges, and we use an algorithm to infer labels for other edges based on the partially labeled initial set. We then use the learned model to infer labels on the held-out test set.

While in general this graph is an incomplete sample from the underlying distribution, we assume that all nodes and edges edges of interest are present in our sample.

Our example has two feature sets, each of which contain exactly one feature. More generally, this bipartite graph representation has a set of features in each feature set. Blum and Mitchell [1] described their data in graph-theoretic terms, by splitting the feature set into two redundant sets. For an example $x_i =< x_{i_1}..x_{i_k}, x_{i_{k+1}}..x_{i_n} >$, we view the instantiations of the features $< x_{i_1}..x_{i_k} >$ to be a one node in the graph, and the example's second set of features $< x_{i_{k+1}}..x_{i_n} >$ to be a second node. The cooccurrence of these features is an edge in the graph, that is, an example is represented by an edge in the graph. We then have an edge in the graph for each example in our data, and a node for each unique instantiation of a set of features. This forms a bipartite graph, since the two feature sets are distinct, and each example instantiates both sets. Nigam and Ghani [8] formed similar bipartite graphs with their data by dividing their feature set randomly into two sets, and showed improvements on semi-supervised learning by using this feature set split.

3.2 Unipartite Graphs over all Example Features

More generally, if we consider each unique feature instantiation as a node in the graph, then an edge between two nodes represents cooccurrence between the two feature instantiations, and an example consists of the set of nodes which are its feature values, and the fully connected graph over them.

4. SMALL WORLD GRAPH PROPERTIES

Small-world graphs have several properties that distinguish them from random graphs. In this section we discuss these properties, and highlight how we can expect algorithms to be affected when the data they are run on exhibit these properties.

4.1 Node Degree

In small world graphs the distribution of node degrees follows a power law.

$$p_k \sim ck^{-\alpha} \tag{1}$$

This means that most nodes are connected to few other nodes, while a few nodes are connected to a large number of other nodes. If our data has this property, then we might expect the degree of the nodes representing our labeled examples to be of some importance in predicting algorithm effectiveness on a data set. It could affect the propagation both of accuracies and inaccuracies in the model. For example, in the cotraining setting, if we correctly label a high-degree node, we obtain correct labels for many different adjacent nodes. Many different examples sharing one half part of the split feature set can be labeled correctly via this node. Conversely, an incorrect label on a high-degree node can propagate the error to many other examples.

This simple observation suggests a hybrid approach to semi-supervised learning that has not been proposed previously, in which greater confidence is required for labeling high-degree nodes than low-degree nodes.

4.2 Clustering Coefficient

Intuitively, the clustering coefficient measures how densely connected the graph is, by measuring, over all connected triples of nodes, how many form triangles. In our semi-supervised learning setting, the clustering coefficient will be useful in predicting the redundancy of features, as well as whether a small sample is sufficiently large to exhibit the true underlying connectivity structure.

Newman and Park [6] show that the value of α from the power law is also predictive of the clustering coefficient. In particular, for $\alpha < \frac{7}{3}$, we expect to see large values of the clustering coefficient C, as C increases with increasing system size.

4.3 Connected Components

The connectivity of the cooccurrence graph is key to the success of any bootstrapping algorithm. Since we hope to learn about a phrase from its cooccurrences, and our algorithms transmit information about likelihood of class membership through cooccurrence links, we are dependent on the existence of links between portions of the graphs which have labels on the edges, and portions of the graphs which have no labels on the edges. In Figure 1, the portion of the graph which contains "<leader, $< x >$ in its industry>" is a separate component. We have no labeled edges in this component.

4.4 Graph Connectivity and Initialization Conditions

We can propagate label information only through edges on the graph. In particular, we cannot propagate label information from one component of the graph to another disconnected component. In Figure 1, we cannot use labels from other portions of the graph to learn to label the edge in the disconnected component which contains "<leader, $< x >$ in its industry>". Thus our set of initial examples and their distribution over components in the graph will be key in our how effective the semi-supervised learning algorithm can be.

4.5 Graph Connectivity and Active Learning

The connectivity of the graph may also explain the importance of active learning for algorithm effectiveness. Active learning may compensate for the lack of component coverage in initial examples.

5. SPEARMAN RANK CORRELATION TEST

To understand the extent to which a variety of experimental conditions predict performance, we can consolidate results from multiple experiments, and see if general trends emerge. By focusing on different properties of experiments, such as the number of examples labeled by seeds, or the number of examples labeled during active learning, and aggregating over multiple experiments, we can measure the degree to which each property affects the results. While each individual experimental result is affected by the combination of conditions, over many different experiments we can see general trends.

To measure these trends, we can perform the Spearman rank correlation test over the results of the experiments, in combination with a candidate predictive property of the experiments. The Spearman rank correlation test is a non-parametric test, *ie* it does not make assumptions about the form of the relationship between two variables. For example, in this chapter we will use the Spearman rank correlation test to test whether the *rank* of algorithm performance is predicted by the *rank* of the number of examples labeled. This means that we can detect a positive relationship between algorithm performance and number of examples labeled with active learning, without making any assumptions about the form of that relationship (for example, without assuming the relationship is linear). The Spearman rank correlation test is related to the Pearson correlation test, but uses the rank of the value rather than the value itself. For our example of measuring how the number of examples labeled with active learning predicts performance, we will order the number of examples labeled, such that each experiment has a rank in that ordering, and order the results, such that each experiment has a position in the ordering of results. Then for all n experiments, the i^{th} experiment gives the pair of $< breakevenScore_i, numExamplesActivelyLabeled_i >$. For each experiment i we find both the rank by breakevenScore: $R_i = rank(breakeven_i)$ and the rank by number of examples labeled with active learning: $S_i = rank(examplesLabeled_i)$. Ties are assigned their average rank. The formula for the Spearman rank correlation test is then found by using ranks in the Pearson linear correlation formula (and is given in Equation 2 [9]). A Spearman correlation score r_s close to 1.0 shows a positive correlation in the ranks. A Spearman correlation score close to -1.0 shows a negative correlation, while scores close to 0 show little correlation. When we calculate the Spearman correlation score we can also calculate the significance level of the test. A significance score of near 0 shows that our measurement of the correlation is statistically significant. Typically we would like to see significance scores of below 0.05 to have confidence in the correlation score.

$$r_s = \frac{\sum_i (R_i - \overline{R_i})(S_i - \overline{S_i})}{\sqrt{\sum_i (R_i - \overline{R_i})^2}\sqrt{\sum_i (S_i - \overline{S_i})^2}} \quad (2)$$

6. SMALL WORLD NATURE OF NOUN-PHRASE CONTEXT COOCCURRENCE GRAPH

We can view our data consisting of pairs $< n, c >$ of noun-phrases and contexts as a graph, if we represent each noun-phrase and each context as a node, and each pair as an edge in the graph. The bootstrapping algorithms we explore exploit cooccurrence information to propagate evidence of class membership. Thus examining the graph structure of the data may provide insight into the expected effectiveness of algorithms on the data. In addition, the extent to which nodes are connected into many or few components will affect the likely performance of algorithms, since cooccurrence information will provide evidence only among nodes in the same component. The presence of seeds in different components may provide insight into the performance of bootstrapping algorithms with a given seed set. Any tendency of active learning to pick out examples in different components may explain how active learning contributes to bootstrapping over data initially labeled with seeds.

If our noun-phrase context pairs exhibit small-world structure, and pronouns and very common nouns form the hubs of the connectivity graph, it may explain the importance of stopwords and frequent seeds in our models.

To see whether our data exhibits small-world structure, we must measure the average node degree, as well as the clustering coefficient which have different values in small-world graphs than random graphs.

6.1 Average Node Degree

When we look at just the noun-phrases in our corpus, we see in Figure 2 that the distribution of the number of contexts they are linked to follows a power law. In Table 4 we see the noun-phrases with the highest degree, which are connected to the most different contexts, *ie* these are the *hubs* of the graph. Note that some of these examples, such as "company" and "customers" are common nouns which are members of our target classes. Others are pronouns which would also be members of our target classes; "he" for example is a member of the `people` class.

Figure 2 also shows the distribution of outdegrees for contexts. Table 4 also shows the contexts with the highest degree. This list contains a mixture of very ambiguous contexts, like "including", which could occur with almost any noun-phrase, and quite unambiguous ones, like "said" which would occur primarily with people and occasionally with or-

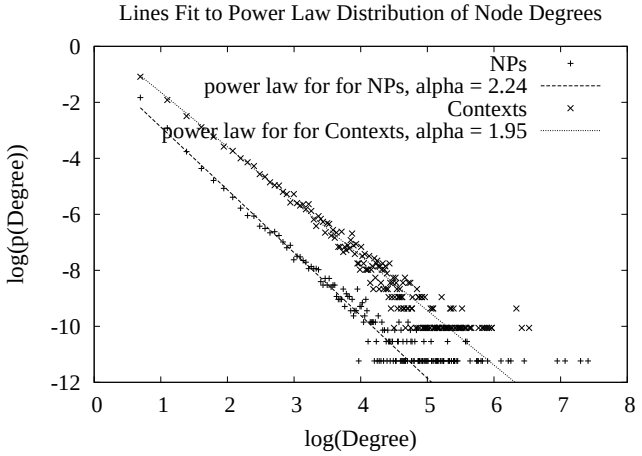

Figure 2: When we fit a line to the log-log plot, we find the power law parameter α is 2.24 and 1.95 for noun-phrases and contexts respectively.

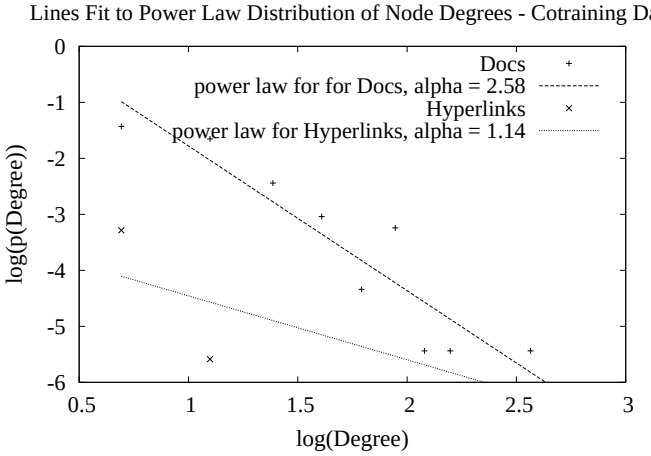

Figure 3: When we fit a line to the log-log plot of the cotraining data, we find the power law parameter α is 2.58 and 1.14 for Docs and Hyperlinks respectively.

ganizations. While some of the highest degree contexts appear similar to one another, for example "$< x >$ provides", "$< x >$ provide" and "`provide` $< x >$', as discussed in Section 2 these provide power for distinguishing distinct cases, particularly whether the noun-phrase is the subject or object of the verb "provide". Subjects of the verb `provide` tend to be agents, such as people and organizations, whereas objects of the verb `provide` tend to be resources.

We can find the coefficient of the power law, by fitting a line to the log-log graph. We have the probability of a node having k neighbors given by the formula

$$p_k = ck^{-\alpha} \quad (3)$$
$$\log(p_k) = \log(ck^{-\alpha}) \quad (4)$$
$$\log(p_k) = \log(c) - \alpha\log(k) \quad (5)$$

Then $-\alpha$ is the slope of a line we fit to the data points.

For contexts, the coefficient of the power law α is 1.95, *ie* we can express the formula for the number of noun-phrases for each context as

$$p_k \sim k^{-1.95} \quad (6)$$

while for noun-phrases, the constant α in the power law is 2.24, *ie*

$$p_k \sim k^{-2.24} \quad (7)$$

Figure 2 shows the lines we fit to the graph of node degrees in the noun-phrase context graph.

The mean degree of noun-phrases is 2.32, while the mean degree of contexts is 7.56. This means that class information about a noun-phrase can be propagated to just over two different contexts on average, while class information about a context can be propagated to over seven different noun-phrases. Labeling a noun-phrase in isolation will affect less nodes than labeling a context in isolation. We also can expect sources of information about the label of a context, since on average seven different noun-phrases will be connected to it, providing more information than the two contexts connected on average to a randomly selected noun-phrase. This suggests that a bootstrapping algorithm such as metabootstrapping, which labels all noun-phrases cooccurring with a given context, will propagate information, or noise, quickly throughout the graph.

In Figure 3 we see a plot of the node degree of Blum and Mitchell's cotraining data, under the assumption that a hyperlink is defined by a unique set of words, and a document is defined by a unique hyperlink. We find that this data also follows a power law.

6.2 Clustering Coefficient

To see whether our data exhibits the small-world property, we will examine the clustering coefficient [7]. Intuitively, the clustering coefficient measures how densely connected the graph is, by measuring, over all connected triples of nodes, how many form triangles. The formula for the clustering coefficient is given by:

$$C = \frac{3 \times number of triangles in the graph}{number of connected triples of vertices} \quad (8)$$

On the training data, we find that the clustering coefficient is 0.22, while on the test data the clustering coefficient is 0.97. This shows that the larger training set has less densely connected nodes, probably because there are few isolated components.

Newman et al calculate clustering coefficients on bipartite graphs representing social networks [7] and found that company directors (a small graph) have a clustering coefficient

Noun-phrase	Outdegree
you	1656
we	1479
it	1173
company	1043
this	635
all	520
they	500
information	448
us	367
any	339
products	332
i	319
site	314
one	311
1996	282
he	269
customers	269
these	263
them	263
time	234

Context	Outdegree
<x> including	683
including <x>	612
<x> provides	565
provides <x>	565
provide <x>	390
<x> include	389
include <x>	375
<x> provide	364
one of <x>	354
<x> made	345
<x> offers	338
offers <x>	320
<x> said	287
<x> used	283
includes <x>	279
provide <x>	266
use <x>	263
like <x>	260
variety of <x>	252
<x> includes	250

Table 4: The twenty noun-phrases and contexts with the highest outdegree. The outdegree is the number of different contexts that the noun-phrase cooccurs with. The noun-phrase list contains a mixture of pronouns, anaphora and common nouns. The context list contains a mixture of very ambiguous contexts, like "including", which could occur with almost any noun-phrase, and quite unambiguous ones, like "said" which would occur primarily with people or organizations.

of 0.59, while movie actors (a larger graph) have a clustering coefficient of 0.199.

6.3 Connected Components

6.3.1 *Measured Graph Components*

Measuring the connectivity of our training corpus, we find 1945 separate connected components. 92129 nodes of 99014 are in the largest component, *ie* 93% of all nodes are connected. However, this leaves 7% of nodes which are not part of the large connected component. The second largest component contains only 107 nodes, with most components containing less than 10 nodes.

The cotraining graph, by contrast, has 189 separate components for 746 nodes, with 91 nodes in the largest component, and 11 nodes in the second largest component. This means that only 14% of nodes are in the two largest components. If all training examples appear only in the two largest components, we will not be able to learn labels for edges for the majority of examples.

6.3.2 *Graph Connectivity and Seed Frequency*

Recall that we initialize our bootstrapping algorithms with a small set of seed words, which are examples of the target class. We could conjecture that seed set frequency will be more important if a graph consists of many unconnected components, and the seeds occur in the largest connected component, or many different components. We observed that frequently occurring examples are important to algorithm effectiveness. Thinking of our data now in small-world graph-theoretic terms, we can see that frequently occurring terms are more likely to be hubs, and are more likely to be connected to many other examples.

We find that for the basic seed sets for the classes `people`, `locations` and `organizations`, given in Table 1, all 10 seeds can be found in the main connected component of the graph. Thus difference between the performance of algorithms over these tasks cannot be explained by differing presence of the main connected component.

For the random sets of country names, more varied distribution can be found in the training set. We showed in earlier work [3] that different sets of seeds have quite a large variance in the number of examples they label in the training set, both in absolute numbers of examples, and in the number of unique examples. We note that the number of unique examples labeled by the seeds is exactly the sum of node degrees of the noun-phrases labeled, since each unique example labeled corresponds to one edge from a noun-phrase to a context. We will first consider how predictive this is of algorithm performance, then contrast it with the number of seeds found in the large connected component.

Figure 4 shows that the total node degree of examples labeled by seeds is more predictive of algorithm performance than is the absolute total number of examples labeled by the seeds. Taking the Spearman rank correlation between these predictors and breakeven score, we find that the total node degree has a correlation score of 0.93, while the total number of examples labeled has a correlation score of 0.92.

Now we examine the number of seeds contained in the largest component for the random country seeds sets. Figure 5 shows that while the number of seeds in the largest component is predictive of algorithm performance (Spearman rank correlation of 0.75), it is much less predictive than the total node degree of examples labeled, which as discussed above has a Spearman rank correlation of 0.93 with final algorithm breakeven score on the test set.

Next we may ask if the number of components with examples labeled by seeds is predictive of performance. Bootstrapping can only propagate labels to other examples in the same

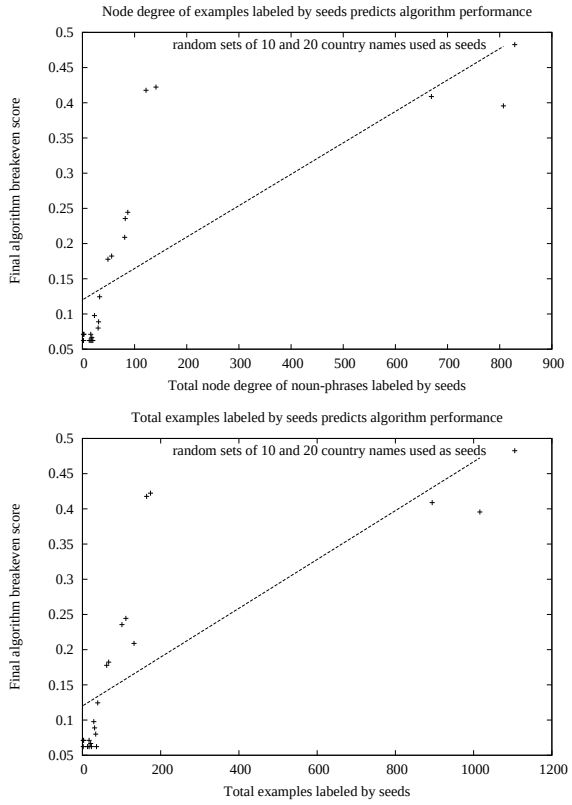

Figure 4: Total node degree of examples labeled with seeds (above, Spearman rank correlation score of 0.93) is slightly more predictive of performance than the total number of examples labeled (below, Spearman rank correlation score of 0.92).

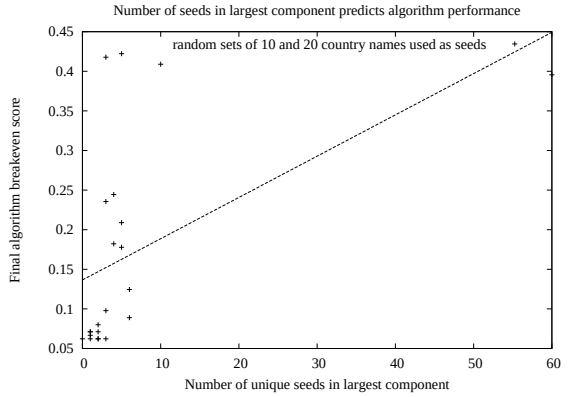

Figure 5: Number of seeds in the largest component is predictive of algorithm performance (Spearman rank correlation score of 0.75) but less predictive than total node degree, shown above (Figure 4).

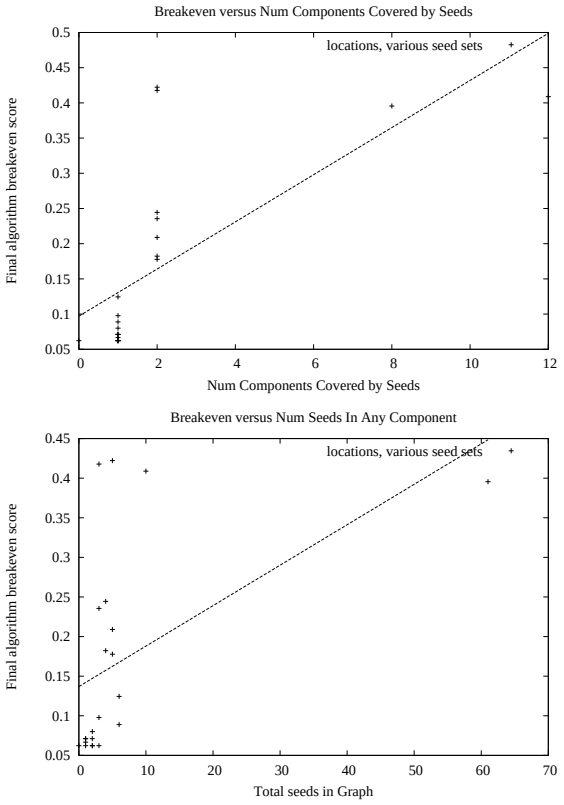

Figure 6: We find that the number of seeds in any component (above) is not as predictive of performance (Spearman rank correlation of 0.75) is just as predictive of algorithm performance as the number of seeds in the largest component, but less predictive than the number of components covered by seeds (below, Spearman rank correlation of 0.87)

component. Recall, however, from Section 6.3.1 that only 7% of examples were found outside the largest component, so seeds labeling examples in the largest component may be sufficient. Figure 6 shows the breakeven score against the number of components covered by seeds in the `locations` class, once again using a variety of sets of country names as seeds. We find that the number of seeds in any component (left) is not as predictive of performance (Spearman rank correlation of 0.75) is just as predictive of algorithm performance as the number of seeds in the largest component, but less predictive than the number of components covered by seeds (right, Spearman rank correlation of 0.87).

Overall we find that the total node degree of examples labeled is more predictive of algorithm performance than the number of components we find seeds in, or the number of seeds we find in the largest component. These comparisons have all been over the locations class. We will now measure the predictiveness of node degree of algorithm performance, across the three classes `locations`, `people` and `organizations`. We see in Figure 7 that node degree is predictive of performance, even across classes. The number of datapoints here is too few to calculate Spearman rank correlation meaningfully.

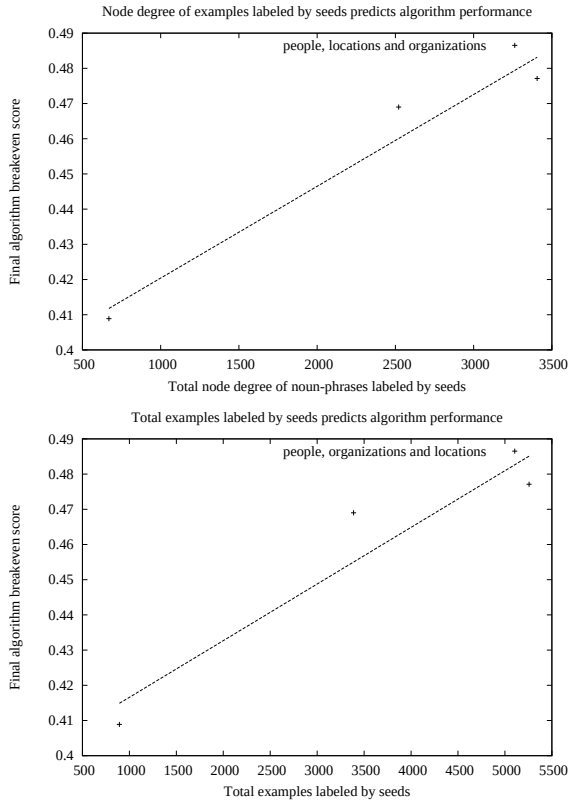

Figure 7: Node degree of examples labeled with seeds versus breakeven, across classes (above) and total number of examples labeled with seeds versus breakeven (below). We find that even across classes, the total node degree of examples labeled with seeds is predictive of algorithm performance.

7. ALGORITHM DESIDERATA FOR SMALL WORLDS

On the basis of the arguments and empirical results presented in this paper, we suggest the following for semi-learning on data exhibiting small-world properties:

- algorithm sensity to node degree: a hybrid approach, in which greater confidence is required for labeling high-degree nodes than low-degree nodes
- initial examples selected to have high degree
- initial examples selected to span many components of the graph
- examples selected for active learning chosen for high degree
- examples selected for active learning chosen for spanning components of the graph
- node degree used as part of feature set selection criteria

8. CONCLUSIONS

We have laid out a set of properties of data sets which we should examine when applying semi-supervised learning, together with a description of their likely impact on learning performance. We have shown that two real world data sets which have been explored in the context of semi-supervised learning exhibit small-world graph properties, and measured algorithm performance in terms of these graph-theoretic properties, showing that node degree of initial examples is predictive of algorithm performance, and that the distribution of labeled examples over components in the graph is also predictive of performance. We also suggested some ways in which algorithms and labeling might be customized around these properties, opening up a range of algorithm design and application that is sensitive to the underlying distribution of the data.

9. REFERENCES

[1] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT '98*, 1998.

[2] R. Ferrer i Cancho and R. V. Solé. The small world of human language. *Proceedings of the Royal Society of London*, B(268):2261–2265, 2001. http://complex.upf.es/ ricard/SWPRS.pdf.

[3] R. Ghani and R. Jones. A comparison of efficacy and assumptions of bootstrapping algorithms for training information extraction systems. In *Proceedings of the LREC 2002 Workshop on Linguistic Knowledge Acquisition and Representation: Bootstrapping Annotated Language Data*, 2002.

[4] R. Jones, R. Ghani, T. Mitchell, and E. Riloff. Active learning with multiple-view feature sets. In *Proceedings of the ATEM workshop*, 2003.

[5] I. Muslea, S. Minton, and C. A. Knoblock. Selective sampling with redundant views. In *AAAI/IAAI*, 2000.

[6] M. E. J. Newman and J. Park. Why social networks are different from other types of networks. *Physics Review E*, 2003.

[7] M. E. J. Newman, D. J. Watts, and S. H. Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences of the USA*, 99:2566–2572, 2002. http://www-personal.umich.edu/ mejn/papers/nasproc.ps.

[8] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *Ninth International Conference on Information and Knowledge Management (CIKM)*, pages 86–93, 2000.

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, second edition edition, 1992.

[10] E. Riloff and R. Jones. Learning Dictionaries for Information Extraction Using Multi-level Boot-strapping. pages 1044–1049. The AAAI Press/MIT Press, 1999.

[11] M. Sigman and G. A. Cecchi. The global organization of the wordnet lexicon. *Proceedings of the National Academy of Sciences of the USA*, 99(3):1742–1747, February 2002. http://www.pnas.org/cgi/reprint/99/3/1742.pdf.