# Determining Topics in Link Traversals through Graph Based Association Modeling

Shelby Thomas
Computer Engineering
UC San Diego
sht005@eng.ucsd.edu

Moein Khazraee
Computer Engineering
UC San Diego
mkhazrae@eng.ucsd.edu

## ABSTRACT

Information networks have long been a technique to understand how a subject moves through a complex maze of information. May it be a user on a computer, a packet in a network, or hub on a street, these networks provide us with valuable information about where a subject is going and why it is going there. In this paper we begin at looking at two such networks. One is a network of citation graphs taken from the high energy physics citations from ArXiv. The second dataset involves a novel game, Wikispeedia in which a user is given a starting and ending node and has to find the shortest path between the two. We compare and contrast these two networks and provide an indepth analysis of both.

We conclude that one of the datasets are more conducive to a predictive task in which we can begin to learn about human understanding. The directed path graph in the WIkispeedia dataset paths lends itself well to tree structures we explore two distinct tree based models and assess the strengths and weakness of these models. In addition we perform a brief runtime analysis and provide ways to optimize these models for large datasets. While it is important to look at the data and find correlations and relationships, we show that it is just as important to understand the story the raw data tell us and set expectations about what can and can't be assumed. Correlation doesn't imply causation and by the end of the paper we begin to see why this is the case.

## 1. DATASET SELECTION

The project began with the evaluation of two separate network based datasets. Here we describe the two networks that we evaluated for the project and what motivated our selection of one over the other.

### 1.1 ArXiv Citation Graph

ArXiv HEP-TH is a citation graph based on papers from high energy physical theory. The data was gathered from the e-print website arXiv and spans the dates from January 1993 to April 2003, a total of 124 months[6][4]. The dataset namely contains a list of 352542 edges and 27770 nodes in the format x y indicating that paper x cited paper y. In addition for each paper we are provided with the time the paper was submitted to arvix. The dataset also contains meta information about the paper which includes its citation index, the author, date, email, and a brief summary of the paper. Our preliminary analysis of the dataset began with looking at graph of complete data set, Fig.1. There is a large community in the center and some small communities around. First of all we wanted to make sure if the center community is a single large community. We wanted to find the importance of papers which resulted to this community, was it seminal, and did it touch many physics sub fields? Moreover, we wanted to find out more about those smaller communities and sub-areas in high energy physics such as particle physics, quatam field theory, or unparticle physics. In Fig.2 edge of the graph is enlarged which shows some of these communities.

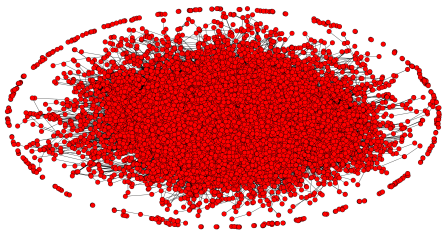

**Figure 1: graph of complete data set**

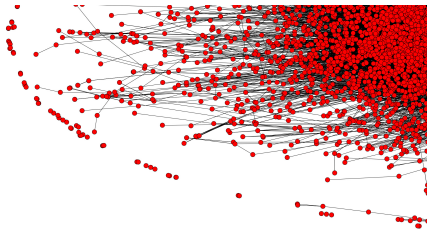

**Figure 2: an edge of the graph of complete data set**

#### *1.1.1 Clique Analysis*

We utilized a clique percolation method by finding cliques of 3 and 4 nodes. After extracting these cliques, we found out that there are 1480565 cliques with 3 nodes and 4129820 cliques with four nodes and the computation is order of $n^2$ with considerable comparison and computation in each iteration. So it was required to select some part of the dataset for our experiment which could be ran in our computers with the limited time. We selected first 60000 edges of the dataset. To make sure our results are reasonable we sketched degree count of both graphs which are shown as Fig.3.

Obviously these selection of data resulted in more nodes with small degree; however, still there are several nodes with high

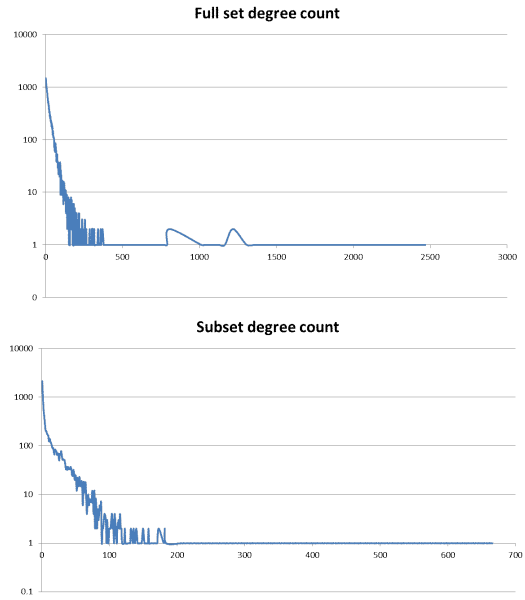


**Figure 3: Degree count for full set and selected subset**

degrees and there are only 70 nodes in the full data set with degree of higher than 300. Hence if we separate the nodes which got out of their community due to these reduction of data set, we can comprehend the results.

Next we ran the community detection algorithm and there were 394 communities of size 4, which meant they did not have any common 3 node clique with other communities. We discarded these nodes since they must be result of subsetting or they may be recent papers which were not cited in this data set. For the rest of the nodes number of communities based on their size is depicted as Fig.4.

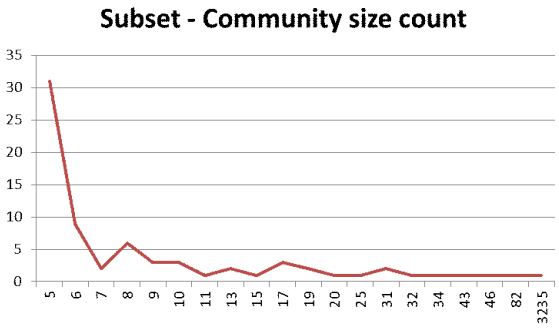


**Figure 4: Number of communities by their size**

We categorized these sizes into 3 classes, very small communities having at most 10 nodes, moderate communities having less than 100 nodes and large communities having more than 100 nodes. As expected there was only one community with high number of nodes, 3235, among total nodes of 8247. There were 485 nodes in the moderate class and the rest in the small one. The interesting point of this analysis is for the moderate class, they must be a subcategory which was explored completely or left due to lack of technology or facilities, or paper from a not well known research center which was cited only locally. Our goal was by using the available metadata and find these moderate fields and predict whether a paper would be highly cited or not.

### *1.1.2 Limitations of Dataset*

Since this data set is for citation, the newer papers would not have been cited and hence would be out of communities. We discarded these nodes by ignoring small communities. Moreover, it had metadata for each paper, including author, email, date of publication, and abstraction. However, it did not have subcategory which would be highly useful for us for the predictive task that we wanted to look. We then took another network graph in the same vein as this one to compare different kinds of direct path graphs.

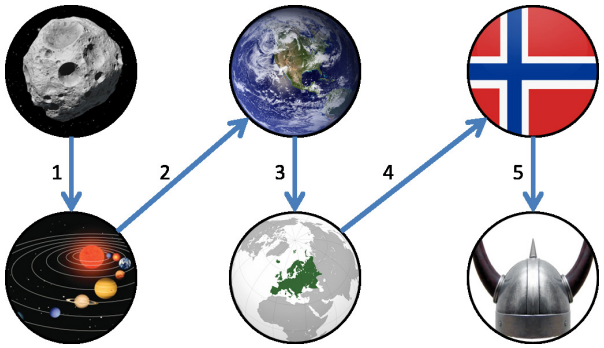


**Figure 5: Graph of a complete path**

## 1.2 Wikispeedia

Wikispeedia is a network of navigation paths based on a human-computation game involving traversals of Wikipedia hyperlinks[18]. The game involves a single player given two distinct links. The goal, as shown in Fig.5 is to find a path between link x and link y. In addition to keeping track of the path between two links, the program also keeps track of time, difficulty, and if the task was completed or abandoned (either timed out or reset). Another interesting feature that the program keeps track of is if the user backtracks. In the path this is associated with a "<" symbol to indicate the user has moved backwards in the path. At the time of release the dataset contained 51,318 finished paths, 24,875 unfinished paths. This results in a network with 119,882 edges and 4604 nodes.

| | Average Click | Average Time |
|---|---|---|
| Finish Path | 4.72 | 158.27 |
| Finished Path Back | 6.75 | 158.31 |
| Unfinished Path | 2.97 | 835.29 |
| Unfinished Path Back | 5.2 | 836.00 |

**Table 1: Path and Time Analysis of the Dataset**

First we aggregated the length of different paths in both the completed and incomplete path graphs. As seen in Fig.6 there was a large difference in the average path length between the people who dropped out in the middle of their run and the completed graphs. Although the average shortest path for all given paths was found to be three links we

found that on average a player who did not go back averaged 4.72 clicks, a player who did go back took 6.75 clicks, and the players who ended up dropping out due to a timeout or reset had an average of 2.97 without black clicks and 5.2 with black clicks. We note high variance in each of these path lengths and dug further by looking at the average time taken for different paths. When looking at the times in aggregate between the three classes of paths, we found a similar trend of low average high variance in all the times. To get a better picture of the time variance in different path lengths we broke down the data by path length and aggregated the time for each. We found that the times in each path are far more consistent but were slightly concerned by the fact that it was difficult to find any correlation between the length of the path, the efficiency of a given search, and the time it took to finish the path. Some of this could be attributed to nature of how the game is played. The two links that are chosen are inherently random so finding the path between certain graphs are far more difficult than others. However looking at table 1 we see that this may not be the case given what we know about the path search times. Although various people had the same path lengths, the time spend for finding that path exhibited high variance. This hypothesis is consistent even if we hold the starting points and ending points to be the same. We took a subset of graphs with 100 of the same starting and ending points and found that the variance here was consistent with the variance of the time it took to complete the graph. We attribute this variance to the various range of knowledge and skill of players. And if we don't normalize for the size path the variance can also be attributed to the random selection of the paths, some being more difficult than others.

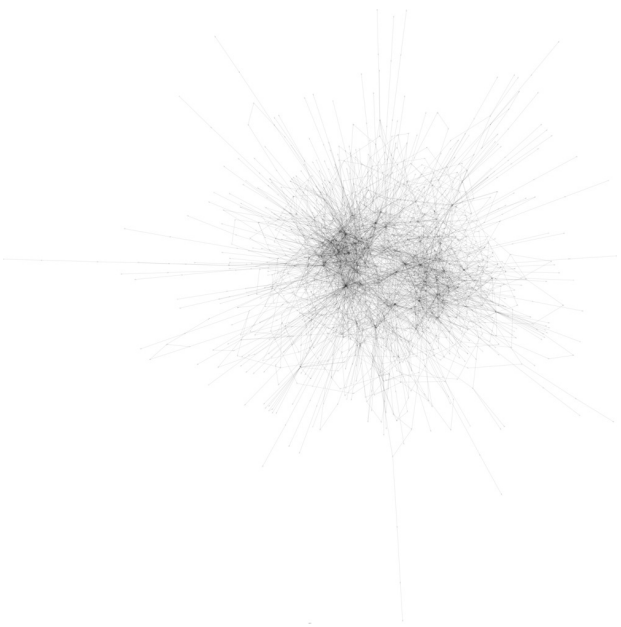

**Figure 7: Graph of 25% of the data with the first and last nodes removed. Even with these common nodes removed there are many dense areas in the graph. The area in the top left side correspond to country names. This can be also be verified by looking at Fig.6. The idea of looking at each community as a possible topic was motivated by this visual interpretation of the data.**

### *1.2.1 Limitations*

One limitation of this dataset is that the paths traversed were selected in a controlled environment and not produced organically. That is to say, the users were directed to a starting and ending point instead of making that decision for themselves. Therefore the part of the graph that we find the most interesting is after the user has passed the first starting point and then begins the organic traversal to the end point. This is a relatively unbiased path as the end point could just be a target that the user had in mind in a real world situation. This forces us to be careful about what kind of information we are gathering from this data and be mindful of the way the data was collected.

## 2. OUT-DEGREE OPTIMALITY

In this section we present our analysis of human behavior in wikispeedia based on given paths. Looking at the paths that performed at the average, optimal, and sub-optimal rate we found various trends in the strategies and patterns in the paths.

### 2.1 Kernel Finding and Path Strategies

One of the most common strategies found was a phenomenon we call kernel finding. This strategy involving going from a starting location finding the most general relationship between the starting and ending points of the given terms. We call these points the graph kernels, where the number of out-degree paths are the most. As a subject becomes more and more specialized the number of out-degree connections decreases as seen in Fig.8 and conversely as the generality of a subject increases the number of out-degree points increase.

We note that this kernel strategy defined in the previous section loosely corresponds to performing a breadth first search on a graph, finding the kernels first and then traversing the graph to become more specific in the search after. This method guarantees a path will be found but not necessarily with the shortest amount of hops possible. We note that the human paths that performed close to optimal had better kernels to go to, resulting in their breadth first search to give them a solution faster than the paths that had a non-optimal kernels. This element of randomness cannot be reconciled however we can try to understand the difference between what kind of graph search method the completed path users took versus the uncompleted path. Now we consider the sub-optimal paths. We define suboptimal paths by looking at the network of unfinished traversals. We take a subset of the paths and remove the first and last nodes. This is done so we do not skew the number of in and out nodes that are presented because of link selection of the game. We see the kernel phenomenon in graphical form here. In fact if we compare the number of out-degrees in the first few links after the first in the finished and unfinished graph we see that on average the number of out-degree connected in the first $n/2$ is much higher in the completed paths. We use this metric based on our analysis of the BFS strategy, the initial nodes that a successful path takes are kernels and out-degrees, that will on average be higher during the first half than the second half.

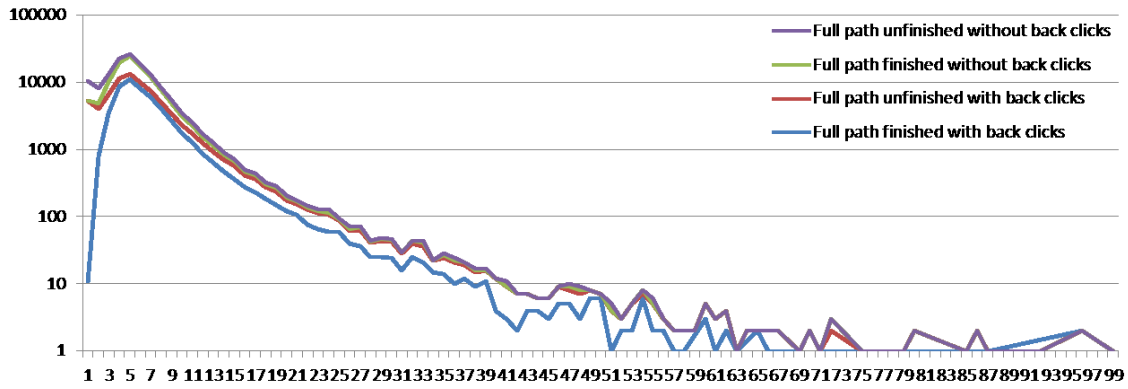


**Figure 6: Click count. The average click count for Full path finished with back clicks, Full path unfinished with back clicks, Full path finished without back clicks and Full path unfinished without back clicks are 6.75, 5.20, 4.72 and 2.97 respectively.**

## 2.2 One Giant Leap

Fig.6 illustrates one of the most discernible points we found through our analysis of the data. When we look at the biggest differences between average path length and sub-optimal path length we find that the quality of the kernels chosen and out-degree is paramount.There is a limitation to the kernel strategy however, in that it is only beneficial to look at kernels during initial rounds of the traversal. In general we found that the starting and end points chosen by Wikispeedia were not kernels based on analysis of end point out-degrees. Most of the end-point nodes were of average or below average degree length. We can begin to construct a picture of how the out-degree value of nodes effects the path chosen. Fig.7 shows that most of the optimal paths follow a triangle-like shape. At the two end points we have the low generality low out-degree nodes, while near the beginning the out-degree value peaks and seem to almost linearly decrease and hit the node at the endpoint. By looking at the point where we start to slope downwards, we see that it is far more difficult to take steps in the correct direction once a kernel has been hit in a path. This is a fact supported by the data from the uncompleted path graphs. We see that people are more likely to give up after they have hit a kernel because at that point the number of out-degree nodes is far less and with each step closer to the target the risk of taking an incorrect step increases. In fact in most paths the longest time is spent after the kernel has reached. Rarely do paths end before hitting a kernel and most of the abandoned paths are due to a wrong "turn" being taken at the midpoint of the path.

Analysing the topics with the high out-degree values became a reprocessing step for us to extra the what would eventually be the output vectors of our model. The top 20 values served as the topic values that we would try to determine given the reverse path graph.

# 3. PREDICTIVE TASK

The predictive task that we are trying to determine is the topic of a node as we traverse from end point to starting point. We present an explanation to the approach we take.

## 3.1 Correlating Associations

The degree of generality in various completed paths provide some information about how paths from node x → y are created but does not tell us anything about how humans search through information to find results. One reason for this is the aforementioned limitation that these data was provided through a highly controlled environment. Instead of looking at the graph from path x → y → z → a, here we discuss the implication of looking at the directed graph through the reverse path, i.e. a → z → y → x.

## 3.2 Analyzing Path Reversals

Our analysis of the generality of the early nodes in section 2 led us to test a hypothesis about finding what people associate when thinking of a given subject. In section 2.2 we stated one strategy for how to successfully complete a path and relies on prior knowledge that the user may have about a certain topic. For example going from solar system to viking through the path Solar System → Planet → Earth → Dutch Language → Netherlands → North Sea → Norway → Denmark → Viking. We see in the beginning the first three nodes, Solar System → Planet → Earth → are far too general to know anything about vikings. However, as the steps become more granular we hypothesize that a better understanding of a given topic leads to shorter paths and higher completion rates. One would have to know that Vikings were affiliated as political entities of Denmark, Norway, Sweden, and Iceland. Initially we saw a linear decrease in the out-degree as we went from the start to end node. Going from the end to the starting node, reversing the path, it follows that we will see a linear increase in the out-degree of the nodes. This follows a bottom-up approach of learning and understanding topics through linearly increasing paths.

### *3.2.1 Limitations of Path Reversals*

One major limitation of the path reversals by our method is that we need to use heuristics about what parts of the path are relevant and what parts are not. Generally going from one side to another should have some sort of reasonable pattern. In contrast this game involved two random end points that may or may not be correlated, so we need to design a way to correct for this inefficiency based on what

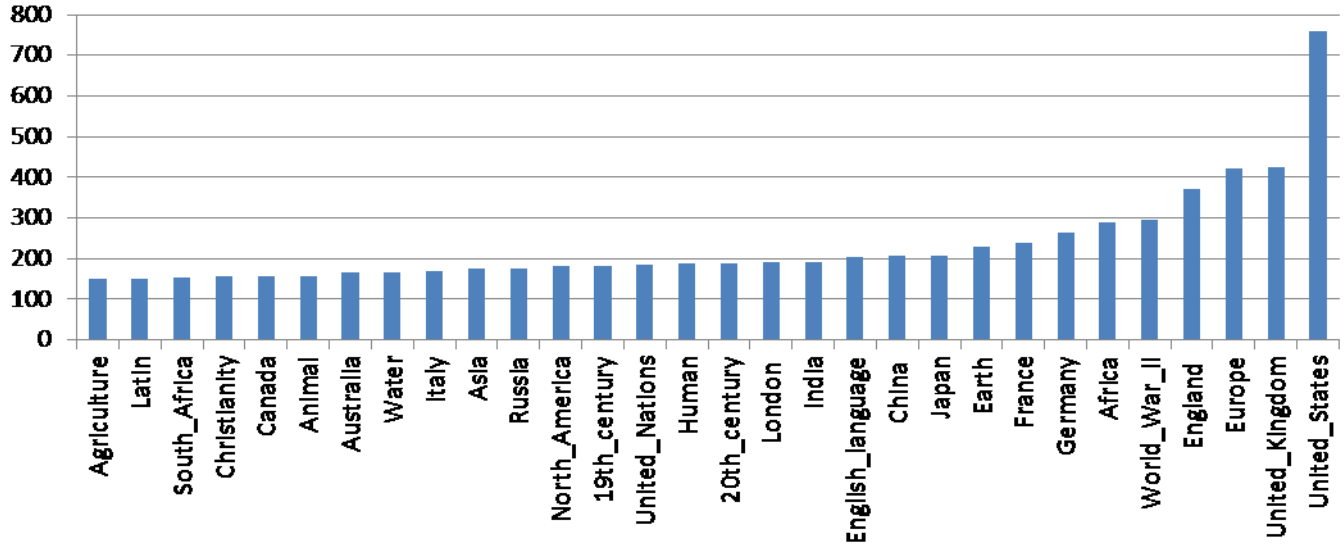


**Figure 8: Links with highest degree**

we have seen about the way the degree of the nodes change.

## 4. RELATED WORK

Trying to find associations and correlations through path graph is a well studied topic that seeks to extract knowledge from site clicks. Srivastava first showed that using a combination of graph preprocessing, pattern discovery and pattern analysis can provide useful data about a graph [15]. Mobasher takes this idea one step further to use this to design better personalization tools [9] for web page and one university in Canada has used path graphs to inference rules from text by creating a dependency tree of a parsed corpus to find rules from text [8], similar to what we were trying to do but their data was more organic.

Much of the other related work in this area is based on the idea of discovering the way links are traversed in order to build a model for improving ads, search, or web sites. West et. al.[17] used the wikipedia dataset to illustrate human way-finding in information networks. The approach that was taken was to create a model to predict where the user is going next based the previously traversed nodes. This paper provides us a basis for us to begin, as some of the conclusion mentioned are confirmed by our work as well. This analysis is an important first pass at trying to understand the data and we take this analysis one step further by changing the structure of the path graph to understand what it means when the path graphs are reversed. The PageRank algorithm which is used to determine the importance and quality of a web page is one seminal example of using out-degrees and in-degrees to create a ranking model[10]. Other derivatives of this including TrustRank[5]. and the Hyperlink-Include Topic Search algorithm came out of this work. Following PageRank, the idea of hubs became of increasing research value as the main factor which determined the quality of a page. If a website has a large number of out-degrees then it would be considered a hub and along with other heuristic to determine value of the content would be scored highly. The hub idea is on that was also reflected in West's paper. They concluded that in a given path from x to y, reaching the hub links in the page as fast as possible is one way to ensure success in path-finding. This was reflected in the data where the computed shortest path between two links found hubs first before switching to specific subjects.

Following click trails has been another focus of commercial for determining if ads have been effective or not. Looking at the trail gives an idea as what topic the user was thinking about at the time of the click and helps improve ad quality (or makes it more intrusive). The paper The Lane's Gifts v. Google Report [16] was an independent study done to determine the way that Google charged for ad clicks. They used two methods to measure this, one was the click-through rate and the other one was the conversion rate. The click through rate specified the links that were clicked and the conversion rate specified people who clicked the ad link and followed through with other actions. Using click trails this way also naturally led to finding compromised computers based on their path traversals. In Eberle et. al.'s paper Anomaly detection in data represented as graphs he notes multiple graph based approach to determine anomalies in graphs. This is done by looking at label modifications, vertex/edge insertions, and vertex/edge deletions [3]. They conclude that by taking a graph based approach to anomaly detection they are able to achieve high detection rate and low false positives with a scalable dataset. Other popular work on spam detection has been done by Levchenko at UC San Diego in a popular paper about click trajectories and the end to end analysis of the spam value chain[7]. In this paper Levchenko and his team build a graph of diverse spam data to determine where the hubs of are for defensive intervention. Similar to PageRank each of these spam hubs is given a value that notes how crucial it is for the entire spam infrastructure.

One aspect that we want to explore is to understand how humans traverse though information and pick out where the

most relevant information will be. On a human behavioral and psychological level, a significant amount of research has been put into determining how humans forage for information. Sellen also takes an inter sting look into how knowledge workers use the web by trying to understand how web sites can be improved based on usage patterns and link traversals[13][11]. They evaluate a large number of people and keep track of their click stream while they traverse the web looking for results. The Priolli paper draw an allegory to animals looking for the sent of a trail to humans looking for the scent of relevant information. Both of these paper mix a cognitive and ecological stance in determining how information is processed. This is invaluable in determining how to design pages and show information in a clear manner so the targets for a user can be reached as efficiently as possible. A recent paper by Scaria et. al. [12] tries to understand why users get frustrated and give up in the middle of their searches. This paper builds on West's paper by looking at the number of times a back click was tracked on Wikispeedia and built a model to try and predict when and where these events happen. The goal here is to design better website for browsing interfaces such that a user will stay on a page to find their goal instead of giving up on navigation.

In our study we look at the click trails in order to determine how humans forage for information however our contribution is that instead of looking in one direction we reverse the path in order to try and build a model for learning. This approach different as we analyze the implications of both forward and backward path traversals. In addition our main contribution is that in order to build a learning model we use associate rule learning to determine relations between broad topics. We leverage several associate rule algorithms including Apriori and ECLAT to find associations between certain topics at the end point of a path. This work can be extended in order to determine what the general public associates with certain people or topics. These may not always be facts, but just what a user perceives associations to be between two topics. This leads to a study of stereotypes or an analysis of the dissemination of false information or propaganda based on phrases users associate with a topic.

# 5. FEATURES

We extract a set of features from the graph based on our initial finding about the data and the relative length of a path, popularity of the kernels, and the relationship between the end nodes and the reverse path phenomenon. One of the feature we look at involves assigning simple weights to the path of the based based on length from the end. This means that the weight of the items moves generally in one direction with the root item being at the end of the node and as the path length increases the length from the end indicates a more general and less useful relationship. For example we take the original path a $\rightarrow$ b $\rightarrow$ c $\rightarrow$ d and reverse it to a $\leftarrow$ b $\leftarrow$ c $\leftarrow$ d. The relationship between c$\leftarrow$d is weighted higher than the relationship between b$\leftarrow\leftarrow$d. In addition we remove all the first nodes of the paths because they are guaranteed to be relatively unrelated to the topic at the end node due to the nature of the Wikispeedia game. Therefore it does not give any useful information and moreover will likely skew the data because there will be a high relationship between common starting and ending points even though both of them have no relationship to each other. The list of features we are evaluating are listed below with the description of their importance.

- Input Vector: Length of the total path graph
- Input Vector: Distance from end node
- Input Vector: Out-degree for each node
- Output Vector: Topic

# 6. MODEL DEVELOPMENT

To evaluate our predictive task we developed two separate models and compare the strengths and weakness of each. First we create a model based on associate rule discovery, a popular data mining algorithm that seeks to find regularities in transaction data. Next we use a kd-tree based model in order to cluster themes together given a certain topic. For the ARD approach we adapt currently well known techniques for association rule discovery to account for the format of our data and the fact that it is a list of sequential paths in reverse order. For both algorithm we take into account the size of our data, perform various optimizations, and present two tree based approaches to finding topic associations.

## 6.1 Association Rule Based Algorithm

We implement our own variant of an algorithm ECLAT which is classified as an Association Rule Discovery (ARD) algorithm which seeks to find regularities in transaction data. We modified the algorithm to find paths that Wikispeedia users commonly traverse through the reverse path graph. While there are many algorithms which are classified as ARD algorithms, ECLAT is unique in that is uses a vertical data format. Generally, data is shown in the horizontal data format, which consists of items which were groups together in specific transactions (TIDs).

The advantage to using the vertical data format is that runtime can be reduced for large datasets. The decrease in runtime occurs because if the transaction ids are in order, it is very easy to find similar transactions between item sets. The larger the itemset, the fewer transactions that need to be processed. If we were to generalize this algorithm to other datasets if the data is given in the horizontal data format, it is necessary to preprocess the data to the vertical data format[1]. Since this only has to be done once, it does not affect the run time of the algorithm. This formatting has a performance advantage that allows us to get results in reasonable time. We explored various data formats because of the large number of edges in our graph. The way we separated edges in the graph was by removing the first node in the forward path and then creating various tuples of edges based on the reverse path.

The second thing we incorporate into our model is the support of an item. The support is the number of transactions that an item or itemset occurs in. It is necessary to keep track of the support of itemsets to show how often the itemset occurs. The higher the support, the more likely the itemset happens. The goal is to find itemsets which meet the minimum support that the user defines. As the algorithm builds itemsets, the support for each subsequent itemset will get smaller until it does not meet the minimum

support. The algorithm recognizes the stop condition when larger itemsets do not meet the minimum support.

The first step of the algorithm is to build a graph which contains all the initial items and the support. This graph is called F2Graph in the algorithm. The second step of the algorithm is to build an inverted database of the vertical data. The goal of this step is to format the data in a way that makes it easy to build two-itemset relations. Using the database each transaction is scanned and the itemset supports are built. In each transaction, combinations of each item are formed and the supports for the item combinations are incremented. The data is gathered after traversing the inverted database and incrementing supports for itemsets. The two-itemsets can be built in a separate graph called EQGraph. The goal of building the two-itemset relations is to expand and build up the n-itemset relations until none meet the support.

The EQGraph displays all two-itemset relations built from the Inverted Database. This graph is used throughout the algorithm to build n-itemset relations. To build the n-itemset relations, a bottom-up search is done to calculate the item intersects based on support. The bottom-up approach a single element from the bottom of the EQGraph is chosen. The n-itemsets are generated by intersecting all neighboring itemsets. For example, if Vikings → Denmark → Europe is chosen as the itemset, then the itemset is represented as {Vikings, Denmark}. The neighboring itemset is {Viking, Scandinavia}. To intersect the two, the support is checked. If any TIDs overlap, then the two create a three itemset and the support is recalculated. Since in our current example, there are no three-itemsets that meet the support of 1, we have met the stop condition of the algorithm.

## 6.2 A Clustering Based Algorithm

In addition we also develop a clustering based algorithm based on the HOP[2] algorithm. We chose to take this approach as it has several advantages over previous algorithms including clearer group assignment, high scalability and a coordinate free implementation making it very flexible in implementation. HOP is applied to some body of N-particles with position and mass indicated via 4 binary floats from an input file. It will cluster the particles into groups based on the most densely populated spaces and assign every particle to a group. This algorithm is guaranteed to converge.

Our algorithm has three main steps, density calculation, the actual hopping, and assignment. Once data has been loaded and initialized, we make 3 passes through the kdtree (the primary data structure). The first pass goes through each particle and calculates that particle density based on the position and masses of its N-nearest neighbors. This density is calculated for Each particle.

A second iteration through the tree is used to determine the densest particles. These particles will serve as the roots of the various groups. This is accomplished by examining the same N-nearest neighbors used in the first pass (again for each particle in the tree). The densest neighbor is then set as the next particle to be examined, and its n-nearest neighbors are checked for highest density. This process continues until the particle being examined, is its own nearest neighbor, i.e. a center for a distinct group. This is done for each particle in the tree until all potential group centers (densest particles) are found.

On the third and final iteration, each particle is assigned to the root of a group (a color). Since most time is spent calculating the density of each particle (for both passes 1 and 2 through the KD tree), the number of resulting groups can be changed (via an argument) and easily reassigned without the need density recalculation [14].

Once the tree is read in, it is sorted and recursively bisected such that each leaf is called a bucket. Each bucket contains a small amount of particles specified via a command line argument. Using buckets allows for simple sorting and quick retrieval of neighboring particles. The compactness of each bucket also contributes to its scalability and low memory overhead. However, after evaluation of the run-time analysis of both of these approaches the clustering approach took an unreasonable amount of time to complete with the 200k+ edges in the graph due to the way hop has to recalculate distances every-time a new node is placed. We were successful in running this algorithm for a fraction of the data

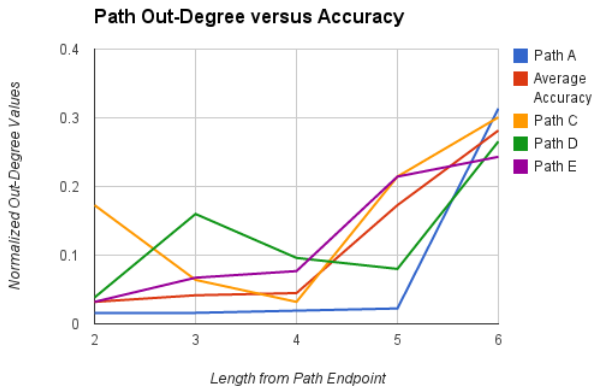


**Figure 9: Links with highest degree**

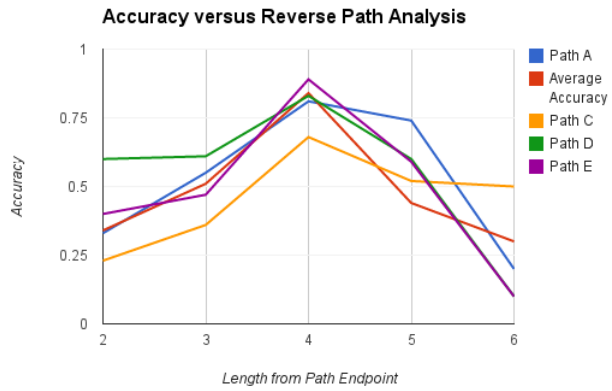


**Figure 10: Links with highest degree**

but ultimately decided to use the ARD approach due to the elegance and speed of the algorithm and the fact that using the clustering approach would add another processing step in order to determine the accurate rates resulting in a fairly inefficient overall algorithm.

# 7. MODEL EVALUATION AND RESULTS

When evaluating our chosen model we found our average accuracy, and then selected five starting nodes in order to see how the algorithm discerns what the subjects or topics of the end node would be. Given the corpus of the common topics, our accuracy rating of the topic changes dramatically as we move toward the end of the path. To show our result we show the average accuracy and the accuracy of five random path of length six as they represent the average path length. Incidentally it also gives the best visual representation of how increasing data does not always increase accuracy.

Due to the nature of the Wikispeedia game where start and end nodes are automatically assigned we wanted to avoid training against false information if there were many repeated paths of unrelated nodes at the edge points. With six nodes we found that generally the first node is completely irrelevant in any way to the final path. This was true for 87.33% of the data and was measured by finding the **inflection point** of the path, i.e. where the out-degree starts to decrease. The rest 13% are paths that were on average less than 4 in length. If we traverse the forward path we note that as the out-degrees increases the data is less relevant to the endpoint, but at the point where the out-degree value decreases we keep that kernel node and every node after.

Next, we reverse the ordering of the path graphs and build edges between the nodes. This reverse association allowed us to create edges that point from a specific topic to a more general one. The end nodes is a specific topic and the ARD model lets us build a predictor as to what associations have strong correlations between each other. Some interesting information we found when training was that there were high correlations between the following topics. United States Cinema $\rightarrow$ Star Wars, Dinosaur $\rightarrow$ Velociraptor, Shark $\rightarrow$ Fish, Carl Friedrich Gauss $\rightarrow$ Denmark. These are four examples of different results when interpreted. One issue of our model is that it tells us fairly obvious relationships, such as the one between Shark and Fish, because the intellectual leap between the two is fairly low. In contract for our second set of results, the one between US Cinema to Star Wars and dinosaur to Velociraptor, it is quite interesting because it allows us to extract topics directly from one hop. Velociraptor to Dinosaur is an example of what happens on the average, as the end point of the node moves toward the front, the out-degrees increase. On the other hand, the US Cinema to Star Wars is an example of moving in this reverse path but with the out-degrees decreasing, because US Cinema is more general than the movie Star Wars. The next class of paths we found were ones that were false positives. For example, Gauss was a German mathematician but a large number of paths associated Denmark with him. We hypothesize this happened because of the small intellectual leap between European countries. Although there was a point in Gauss's life where he did work in Denmark for his geodetic work, we still classify this as error. Class 1 and Class 3 training ended up being the biggest pitfalls of our model. For one, the information would be very obvious and almost not useful. Moreover, there is an error rate that is relatively high because in the second half of the path a user could switch from a low out-degree node to a kernel and then back to a low degree value to get to their target with the kernel being unrelated to the end node.

We have summarized the accuracy of our results in the Fig.9 and Fig.10 based on the average path accuracy and single reverse path accuracy. We note that at the first jump, two, there is a high accuracy rate of what the topic is going to be. After this point, the accuracy peaks at around click 4 on average and then starts to decrease. For the test data we tested against the full path so it would be clear to see where the inflection point is. We see that around the fourth jump, the data starts becoming less relevant to the end node's subject. To verify that this phenomenon was really because of what we hypothesized we also graphed the accuracy of the four randomly chosen paths and the average path to see if it was a correlation between the out-degree values and the length of the endpoint. This was verified and if you lay the second graph on top of the first, it is clear to see a strong relationship supporting our initial guess about why accuracy rates were increasing and peaking at a certain point and then decreasing.

# 8. CONCLUSION

Counter-intuitively, our result was that more data does not necessarily increase the accuracy of our model in this case. In fact to a certain extent the accuracy of our model is inversely proportional to the number of nodes we choos, and by association the number of out-degrees that a node has. Although our model is not perfect, we presented its shortcoming, strength, and show that is possible to determine what people associate with specific word and find the topics based on the data from the Wikispeedia network. We would like to perform further analysis on this data by looking if it is possible to predict the end node based on the forward path. In addition we believe that this class of data is conducive to doing human behavioral research to find what stereotypes exist, how people feel about a certain person or topic, or people's views on a subject. Continuing this research this way requires us to have a larger and more broad dataset of status from Twitter, other social media, and blogs and some data about the way people are traversing these networks. In addition, if we had data that was localized and sorted by region it would provide a great amount of interesting behavioral and political insight. We hope the perspective that we have given on the usefulness of path backtracking will assist in the development of interesting technologies to understand how users utilize online and offline resources to disseminate their opinions and learn new things.

# 9. REFERENCES


[1] C. Borgelt. Efficient implementations of apriori and eclat. In *Proc. 1st IEEE ICDM Workshop on Frequent Item Set Mining Implementations (FIMI 2003, Melbourne, FL). CEUR Workshop Proceedings 90*, page 90, 2003.

[2] M. Chatterjee, S. K. Das, and D. Turgut. Wca: A weighted clustering algorithm for mobile ad hoc networks. *Cluster Computing*, 5(2):193–204, 2002.

[3] W. Eberle and L. Holder. Anomaly detection in data represented as graphs. *Intell. Data Anal.*, 11(6):663–689, Dec. 2007.

[4] J. Gehrke, P. Ginsparg, and J. Kleinberg. Overview of the 2003 kdd cup. *SIGKDD Explor. Newsl.*, 5(2):149–151, Dec. 2003.

[5] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 576–587. VLDB Endowment, 2004.

[6] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 177–187, New York, NY, USA, 2005. ACM.

[7] K. Levchenko, A. Pitsillidis, N. Chachra, B. Enright, M. Félegyházi, C. Grier, T. Halvorson, C. Kanich, C. Kreibich, H. Liu, D. McCoy, N. Weaver, V. Paxson, G. M. Voelker, and S. Savage. Click trajectories: End-to-end analysis of the spam value chain. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, pages 431–446, Washington, DC, USA, 2011. IEEE Computer Society.

[8] D. Lin and P. Pantel. Dirt @sbt@discovery of inference rules from text. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 323–328, New York, NY, USA, 2001. ACM.

[9] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Effective personalization based on association rule discovery from web usage data. In *Proceedings of the 3rd International Workshop on Web Information and Data Management*, WIDM '01, pages 9–15, New York, NY, USA, 2001. ACM.

[10] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web, 1998.

[11] P. Pirolli and S. Card. Information foraging in information access environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 51–58. ACM Press/Addison-Wesley Publishing Co., 1995.

[12] A. T. Scaria, R. M. Philip, R. West, and J. Leskovec. The last click: Why users give up information network navigation. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 213–222. ACM, 2014.

[13] A. J. Sellen, R. Murphy, and K. L. Shaw. How knowledge workers use the web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, pages 227–234, New York, NY, USA, 2002. ACM.

[14] Y. Shi and Y. T. Hou. A distributed optimization algorithm for multi-hop cognitive radio networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE, 2008.

[15] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explor. Newsl.*, 1(2):12–23, Jan. 2000.

[16] A. Tuzhilin. The laneâĂŹs gifts v. google report.

[17] R. West and J. Leskovec. Human wayfinding in information networks. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 619–628, New York, NY, USA, 2012. ACM.

[18] R. West, J. Pineau, and D. Precup. Wikispeedia: An online game for inferring semantic distances between concepts. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 1598–1603, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.