Unifying Clinical Trials and Publications via a Dependent Clustering Approach

Lauren Waldrop
The University of Texas at El Paso
500 West University Avenue
El Paso, Texas 79902
lewaldrop@miners.utep.edu

Wenqing Sun
The University of Texas at El Paso
500 West University Avenue
El Paso, Texas 79902
wsun2@miners.utep.edu

Jessica Rebollosa
The University of Texas at El Paso
500 West University Avenue
El Paso, Texas 79902
rebollosa.jr@gmail.com

Marcus Gutierrez
The University of Texas at El Paso
500 West University Avenue
El Paso, Texas 79902
mgutierrez22@miners.utep.edu

ABSTRACT
Clinicaltrials.gov [1] houses information regarding clinical trials that are currently underway. In addition to information about background, purpose, and design of a specific clinical trial, the webpages also provide links to affiliated papers that can be found in PubMed [2] (a warehouse for citations in biomedical research). These links are explicit, but implicit links between clinical trials and publications more than likely exist. For example, a researcher may like to know if a given clinical trial is related to more publications than just the ones listed on the clinical trial webpage. This relation could be the result of similar key terms imbedded within the clinical trial webpages and PubMed abstracts. By using a dependent clustering algorithm [3], and a novel approach using Naïve Bayes for heterogeneous datasets, we aim to give scientists in the biological community insight not only into related terms, but also clinical trials and/or other publications that may not have explicit links.

Categories and Subject Descriptors

General Terms
Algorithms, Design, Experimentation

Keywords
Clustering

1. INTRODUCTION
Clinical trials introduce novel techniques that prevent, diagnose or treat disease. Often publications are generated as a result of a clinical trial and reveal key information regarding the approach, methodology and results. Clinicaltrials.gov, a database which maintains public information about current ongoing clinical research studies, lists publications that are affiliated with each clinical trial. These same publications can be found in PubMed. Relations between a given clinical trial and its associated publications already exist; however, there may be undiscovered links between a given clinical trial and other publications, or even other clinical trials.

With this study, we are hoping to utilize pre-existing explicit links to provide additional insight with regard to implicit links between clinical trials and biological publications to individuals within the scientific community. In addition to implicit links, we are hoping to reveal similar terminology that exists between linked publications and clinical trials.

2. DATA DESCRIPTION
The data for this project consists of terms found in webpages from Clinicaltrials.gov and abstracts in PubMed. The data is organized into two weighted term document matrices, one for data gathered from Clinicaltrials.gov, and the other from PubMed abstracts. Each matrix row is associated with a single document, while each column is associated with a term found within the document.

3. Data Preprocessing
For this phase of the project, the frequencies of each term are weighted using term frequency-inverse document frequency, more commonly known as tf-idf [4]. Each tf-idf value is increased proportionally with the number of appearances within a single document, but is balanced out by the number of times the term appears within all documents examined. Many TF-IDF thresholds were explored in the generation of our datasest. What we found was that lower thresholds (i.e. 0.01) tended to produce better results for both k-means and the dependent clustering algorithm.

In addition to the two weighted document term matrices, we utilize a relations file that illustrates the explicit relationships between clinical trials and PubMed publications. All three of these documents are used in the dependent clustering algorithm described below.

After reviewing the results section of this paper, it becomes quite obvious that going forward, we will more than likely need additional pre-processing constraints. More specifically, we need to clean-up the data for both PubMed abstracts and Clinical Trials excerpts. To do this we are proposing a two-step algorithm. The first step involves concatenating all of the words of each text file into two files for each dataset – one that contains the names of the files, which will be used to create the tf-idf weighted term matrices, and the second contains just the words that will be used to create the list of terms.

For the generation of weighted document-term matrices in previous assignments, we had used the Textmodeller package, which provides document, term and matrix files in the appropriate format for dependent clustering to use as input. We had noticed that using this method on the pre-processed data generated entities or phrases that in some cases were coherent, but the majority of the entities turned out to be of no real use. For example, Textmodeller did return entities such as “thyroid cancer” or “stem cell”. In both cases, the meaning of those two words together

adds more specificity and context than either one could do on their own. However, the majority of the entities were a concatenation of terms such as the following, "arms arm arm oxaliplatin arm arm". In this example, the combination of those terms together doesn't seem to provide any additional meaning or insight into the phrase. As a result, we decided it was best to run two different weight matrices through the dependent clustering code: one that had been generated via Textmodeller, and the other that was generated from individual terms, such as "adenocarcinoma" or "neoplasia".

Textmodeller codes produced 1,696 unique terms for PubMed documents, and 1,492 unique terms for the Clinical Trials documents. A simple solution for reading documents and finding TF-IDF values for single terms was implemented in Java. This process yielded 818 unique terms for Pubmed and 802 unique terms for the Clinical Trials dataset.

After reviewing the results section of this paper, it becomes quite obvious that going forward, we will more than likely need additional pre-processing constraints. More specifically, we need to clean-up the data for both PubMed abstracts and Clinical Trials excerpts. To do this we are proposing a two-step algorithm. The first step involves concatenating all of the words of each text file into two files for each dataset– one that contains the names of the files, which will be used to create the tf-idf weighted term matrices, and the second contains just the words that will be used to create the list of terms. After the files are generated, punctuation, numbers, and any unwanted symbols such as '#', '%', ':' will be removed from the list, and all letters will be converted to lower case. Next, repeated words will be removed from the list. Our first approach tolumping together similar terms and getting rid of unwanted terms involved both the use of ignore lists and WordNet. However, our datasets still maintain quite a bit of noise. Since we are primarily interested in biological terms, the second step of the algorithm required the use of LingPipe tool kit which include two data models to make a data dictionary. Using the the code provided in the tutorials by LingPipe as base, we trained a Name Entry Recognizer with two data models [6] to get the type of the terms. One of the challenges was defining the type since we needed to keep more than one type of terms. Even though these codes provided a good insight of what the terms should look like, the code results threw sentences instead of words. In order to get better results, we used some code from BeCAS: biomedical concept recognition services and visualization. Which allowed us to get better terms. Here are some of the final terms we used for the Trials and PubMed abstracts: abdomen, abdominal, abg, abt, ace, acetate, acitretin, acth, adenocarcinoma, adenocarcinomas, adenoid, adenoma, adenomas, adenovirus, adjuvant, adriamycin, aggressive, agonist, agonists, agus, air. Once the biological terms were defined, we were able to remove all the unwanted words from the list of terms.

4. METHODOLOGY

In this project, we are taking two different but complementary approaches. The first of which involves a dependent clustering algorithm, developed by Dr. M. Shahriar Hossain in an effort to find implicit relationships and similar terms between the clinical trials dataset, and the PubMed abstracts dataset. The second approach involves the development of a novel technique for classifying documents from heterogeneous datasets via Naïve Bayes. Following is a short description of both approaches.

The first step of the dependent clustering algorithm is to separately assign vectors in each of the two datasets to clusters via k-means. The second step involves preparing contingency tables based on the clustering results and the pre-existing relationships between the two datasets. Finally, each of the contingency tables are evaluated by minimizing a cost function such that relationships in one cluster of the clinical trials dataset are exclusive to only a single cluster in the PubMed dataset. These individual steps are repeated until convergence.

Finally, Naïve Bayes Classification will be applied to the heterogeneous data set. This algorithm uses training data to predict classes of new data entries. In this context, the Naïve Bayes Classification will either reinforce existing links or suggest new links that may better cluster the data and provide insight in to the architecture of the data set. The result may also further advance the data preprocessing phase by adding previously implied links further connecting the two data sets.

5. RESULTS

5.1 Term Elimination via Term Variance

In the initial phases of our work, we wanted to see how our data would cluster using the implementation of dependent clustering that was authored by Dr. M. Shahriar Hossain. Because most of the data were clustered into a big cluster and we decided to remove some not important (the term with less descriptive power) terms to improve the performance. In our last phrase we already tried our methods with the data generated with a TF-IDF cut-off value of 0.3. In this phase, we tested the method on the data with threshold 0.01.

K-means has been shown to work very hard to place roughly the same number of instances in each cluster, and since our preliminary results varied significantly from this school of thought, the number of terms in each dataset was reduced [7]. To do this, terms for both PubMed and Clinical Trials datasets were eliminated by leveraging the variance of each term. Variance is a good way to measure the differential power of a term. Low variance of a term indicates that either the term is not present in any of the documents, or the term is present in most or all of the documents. Thus, terms with low variance do not maintain any discriminatory power.

To rule out terms from both datasets with low variance and exceptionally high variance, document-term matrices were generated, the variance of each term was calculated, the terms and their associated variance values were ranked in decreasing order, and the results were plotted. Figure 1 illustrates the resulting plots for both cases.

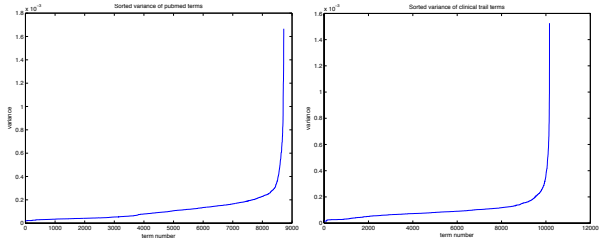

Figure 1. Term Variance Plots for both the PubMed dataset (left) and the Clinical Trials dataset (right).

Using these plots, a set of minimum variance thresholds was established for each dataset. Compared to the method in our last

phase, we didn't set the maximum threshold because the higher variance has better ability to describe the label of the documents. Once the variance thresholds were determined, k-means was run with different minimum thresholds. We used two different measurements to choose the best threshold, one is the term percentage in the largest cluster and the other one is ASC. The figures are shown below. And the thresholds we choose are 0.00008 and 0.000070.

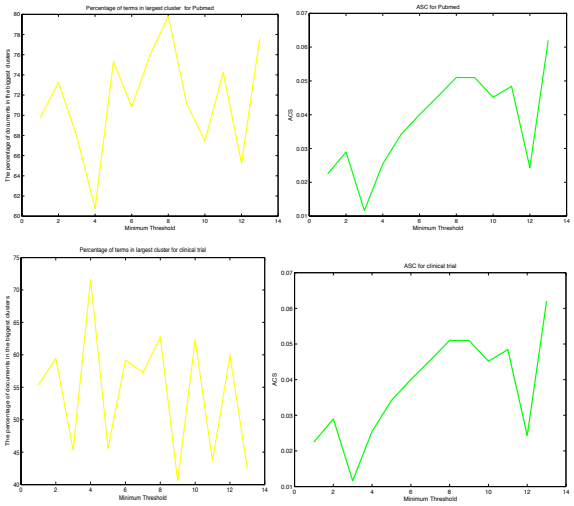

Figure 2. Term percentage in the largest cluster plots (left figures) and ASC plots (right figures) for PubMed and Clinical trial dataset.

5.2 Dependent Clustering

Determining the optimal maximum and minimum thresholds allowed us to reduce the number of terms in both the PubMed and the Clinical Trial datasets. Dependent clustering was run again to determine whether or not feature reduction by variance helped to more equally distribute documents amongst clusters.

Histograms of the clustering results were generated before and after applying dependent clustering on each of the datasets. To test the differences of using each entity as feature and using each term as feature, we applied the dependent clustering algorithm on both datasets. The figures are shown below.

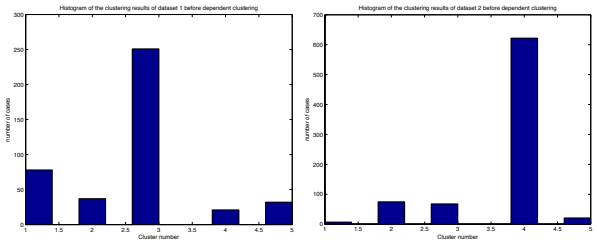

Figure 3. Clinical Trials (left) and PubMed (right) Frequency of Documents in each Cluster before dependent clustering was applied (k=5). Each entity was used as a feature.

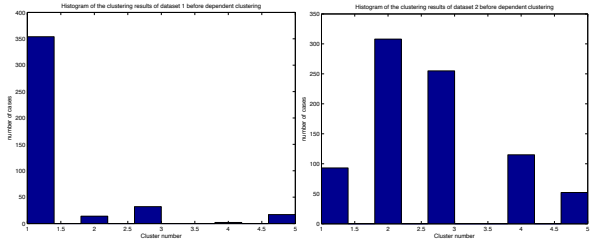

Figure 4. Clinical Trials (left) and PubMed (right) Frequency of Documents in each Cluster before dependent clustering was applied (k=5). Each term was used as a feature.

From the figures we noticed that all of the documents in both datasets were classified into one large cluster. But using entiry or term as feature makes some different on these two datasets: When using the term, the largest cluster for PubMed contains a little bit more than 300 entries, and the largest cluster for Clinical Trial contains about 350 entries. When using the entity as the feature, the number becomes 600 and 250.

For both datasets, after the dependent clustering, we only get one giant cluster for both Clinical Trial and PubMed.

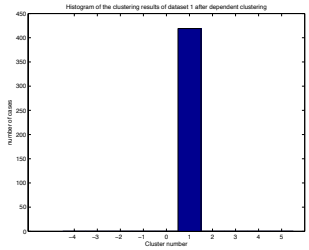

Figure 5. Clinical Trials frequency of documents in each cluster for both datasets after dependent clustering was applied (k=5).

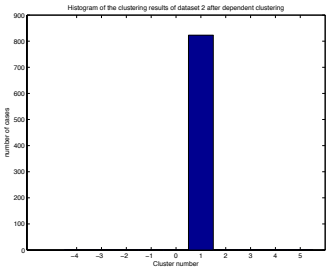

Figure 6. PubMed frequency of documents in each cluster for both datasets after dependent clustering was applied (k=5).

5.3 Data Visualization

To have a better understanding as to why the data is not working, we aimed to obtain a better visualization of the data. Since our data is high dimensional data, meaning that it has thousands of variables, we cannot simply plot out the original data. In this project, we used the principal component analysis (PCA) algorithm to reduce the dimensionality of the original data [8]. With the help of PCA, we obtain the most informative data from the original dataset. After running PCA with two principal components, we obtain PC1 and PC2, which contain the first and second most information from the original data. As such the dimension of the data was reduced to two. Figure8 illustrates the PCA plots for both PubMed and Clinical Trials data. From the figures, we can see no obvious separation of clusters, and most of

the data is clustered into one large group. We also observed several outliers which might explain why dependent clustering gives us one big cluster and several very small clusters. The figures below show the patterns of our PC plots for using entity as feature and each term as feature. And we can see using different features, the PC plot patterns are very different.

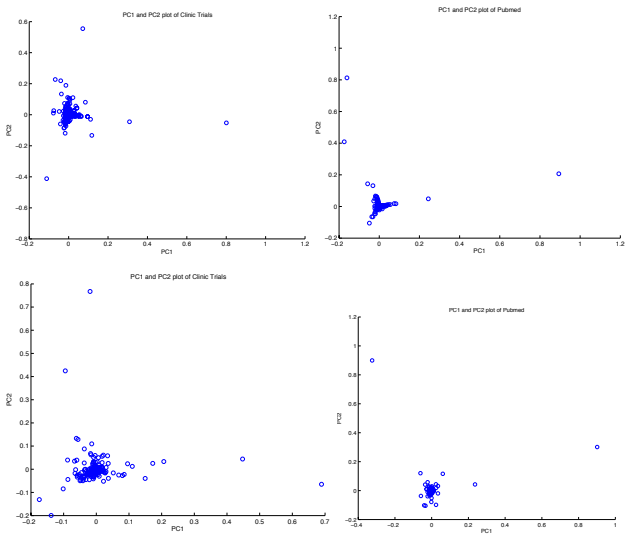

Figure 7. Results of principal component analysis (PCA) applied to both the PubMed and Clinical Trials datasets. The upper figures show the PC plots using each entity as a feature, and lower figures show the PC plot using each term as a feature.

5.4 Term Analysis

To analyze why the dependent clustering is not working, first, we tried to explore more into each cluster, and we found kmeans failed to separate the data into different categories. For example, the first cluster, which is the biggest cluster, contains a stuff including cell, tumor. The second cluster contains kidney, hormone. The third cluster is related to prostate, skin. The third group involves many terms overlapped with other clusters like tumor, cell. And the third group is a very tiny group which also has many terms overlapped with other group, and it has some high frequency word like microcalcification, carcinoma. The data we used contain several different types of cancer, and we hope we can use the clustering algorithm to separate them, but the results are not as good as we expected. One possible reason is some other terms dominated the whole dominated, so the clustering is not good. More terms are shown in the figure below:

Cluster number	Terms
1	Gemcitabine, fluorouracil, pancreatic, adenocarcinomas, tumor, cells, leucovorin, cancer, pancreas, bile
2	Breast, mbc, mtd, leucovorin, cancer, tumor
3	Fenretinide, renal, kidney, oral, interleukin, ifosfamide, cbdca, etoposide, transplant, lymphoma, engraftment
4	Cells, leucovorin, cancer, bcg
5	Neoplasia, selenium, cellular, proliferation, ducts, chromatin, vitamin

Table 1: The overview of the random terms in each cluster.

To have a better understanding of what is inside in biggest cluster, we plotted analyzed the frequent terms. Due to the limitation of the time, all the following analysis was using the data with each term as a feature. There 354 documents in the largest cluster. We plotted the distribution of the terms, and for the clinical trial data, there are 802 terms and the most frequent term has appeared 330 times in this cluster. For PubMed data, it has 823 documents, and the largest cluster has 308 documents with 818 features (terms). The table below shows the top 10 most frequent terms and their frequency.

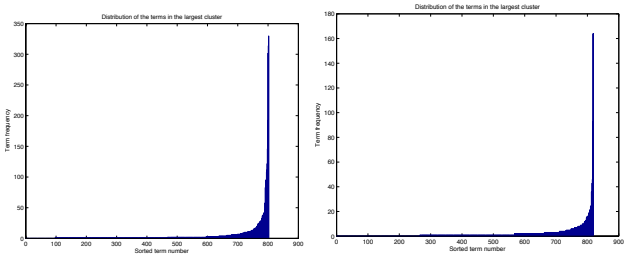

Figure 8: Distribution of the terms in the largest cluster for Clinical Trial data (left) and PubMed data (right).

Clinical Trial		PubMed	
Term	Frequency	Term	Frequency
cancer	330	carcinoma	164
tumor	314	cell	93
cells	302	cancer	46
cell	193	renal	43
carcinoma	146	head	36
arm	122	neck	36
lung	112	patient	25
arms	108	cells	24
patient	96	cisplatin	24
cisplatin	95	cardia	22

Table 2: Top 10 most frequent terms and their frequency in the largest cluster for Clinical Trial data and PubMed data.

Also we also analyzed the second biggest cluster, there are 32 and 255 documents in the second largest cluster for each data. And the figures of the term distributions and table of term frequencies for both Clinical Trial and PubMed are listed below:

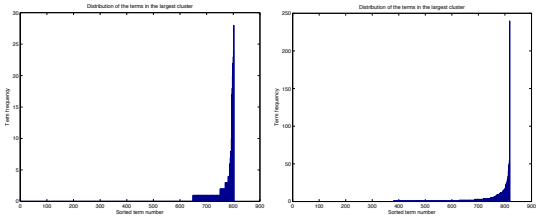

Figure 9: Distribution of the terms in the second largest cluster for Clinical Trial data (left) and PubMed data (right).

Clinical Trial	PubMed

Term	Frequency	Term	Frequency
cancer	28	cancer	240
tumor	24	cisplatin	55
carcinoma	23	paclitaxel	51
cell	22	cell	48
cells	22	lung	42
renal	20	carcinoma	41
kidney	18	adjuvant	34
growth	17	adenocarcinoma	31
blood	16	fluorouracil	29
interferon	10	gemcitabine	29

Table 3: Top 10 most frequent terms and their frequency in the second largest cluster for Clinical Trial data and PubMed data.

For the third largest clusters, there are 17 and 115 documents for each datasets.

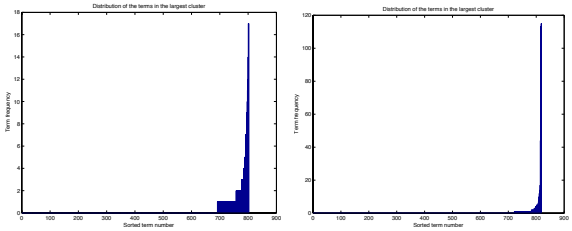

Figure 10: Distribution of the terms in the third largest cluster for Clinical Trial data (left) and PubMed data (right).

Clinical Trial		PubMed	
Term	Frequency	Term	Frequency
prostate	17	lung	115
cancer	17	cancer	113
cells	14	cell	46
tumor	12	nsclc	44
adenocarcinoma	11	paclitaxel	17
arm	10	leukemia	14
psa	9	adjuvant	14
arms	9	carboplatin	12
antigen	7	cisplatin	10
oral	7	cranial	8

Table 4: Top 10 most frequent terms and their frequency in the third largest cluster for Clinical Trial data and PubMed data.

For the rest two clusters (small clusters), we also had a brief look at them and the listed the top three clusters. For clinical trial data, the top three clusters of the fourth the last clusters are: {cancer, breast, women} and {bcg, bladder, recurrence}, respectively. For the PubMed data the top 5 terms of the rest two relatively small clusters are {cancer, ovarian, prostate, cervical, adjuvant} and {breast, cancer, adjuvant, letrozole, aromatase}.

From the results above, we decided to remove the common terms. For Clinical Trial data, we removed {cancer, tumor, cells, cell}, for PubMed data we removed {cell, cancer, cells}. And we set k=8, rhoDivD=0.001, Then we reran the Kmeans clustering, and we get the following results:

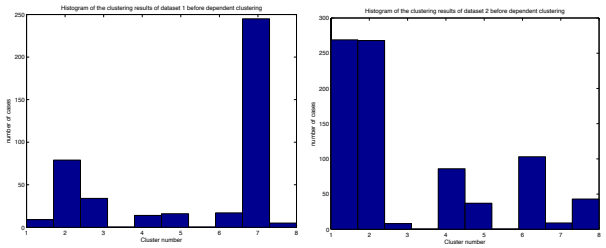

Figure 10: The histogram of clusters before dependent clustering for Clinical Trial and PubMed dataset.

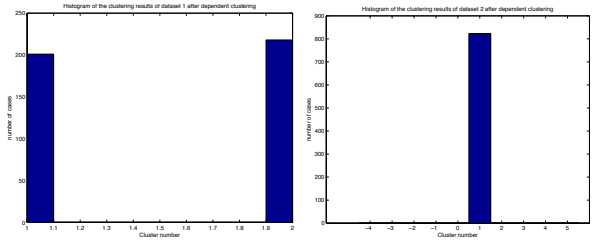

Figure 11: The histogram of clusters after dependent clustering for Clinical Trial and PubMed dataset.

After the dependent clustering, the clinical trial data was divided into two clusters. And we analyzed the top 10 terms and their frequencies in each cluster. There are 201 documents in cluster 1 and documents in cluster 2.

Cluster 1		Cluster 2	
Term	Frequency	Term	Frequency
carcinoma	89	arm	100
lung	64	carcinoma	97
patient	57	arms	92
oral	55	cisplatin	63
arm	43	patient	52
paclitaxel	39	oral	50
arms	36	paclitaxel	50
cisplatin	34	lung	49
blood	34	carboplatin	40
mtd	29	growth	36

Table 5: Top 10 terms of each cluster after dependent clustering

From the table, we found most of the terms in each cluster are overlapped, which means the dependent clustering made the clustering results worse.

In addition, we also tried k=5, 10, rhoDivD=0.01, 0.00001, the dependent clustering is still not working good, only gave us one giant cluster.

6. NAÏVE BAYES CLASSIFICATION

6.1 Motivation

Part of the heterogeneous dataset in question consists of explicit relations between the PubMed articles and the clinical trials. However, each PubMed article has no more than two relations, with the vast majority containing only one explicit relation. This can become problematic as only little information is gained from each article only containing one relation. If an algorithm can

accurately suggest more relations to further define how the two datasets are connected, this should, in theory, improve the dependent clustering algorithm. Naïve Bayes Classification suits perfectly for this task, with the addition of predicting relations to documents that do not exist in the training data.

6.2 Implementation

Naïve Bayes Classification had to be modified to handle the document-to-document analysis. In essence, when determining the probability a document in corpus A is related to a document in corpus B, the sum of all the probabilities of every term in document B, given every term in document A, leading to the following formulas.

$$P(D_B|D_A) \propto \sum_{b=1}^{N_b}(P(t_b) \cdot \prod_{a=1}^{N_a} P(t_a|t_b))$$

$$P(t_b|t_a) \propto \frac{N_A(t_b+1)}{\sum_{i\in A}^{A} N_A(t_i+1)}$$

Where A is the total number of unique terms in document A and $N_A(t_i)$ is the number of times term t appears in document A. Laplace smoothing adds an extra instance of each term in in order to avoid the detriment of a value of 0 to computations. Because these results are proportional to probability, but not actually the probability, we will use the term ratings to refer to the rese results.

After implementation of this algorithm on Java, the output was sent to a text file that contained the rating of every pair of documents from PubMed to Clinical Trials. This large file contained numerous double values approaching zero due to the product of large denominators in calculating $P(t_b|t_a)$. This clearly raised a question of accuracy amongst the team. In traditional Naïve Bayes Classification, logarithms are placed around the probabilities to improve accuracy and speed, allowing for the summation and subtraction of logarithms as opposed to multiplication and division of term counts. However, this same strategy does not hold true for the modified Naïve Bayes Classification. The rating value of traditional Naïve Bayes Classification consists of one term comprised of the product of multiple factors. In comparison, this modified heterogeneous version consists of the sum of multiple terms each comprised of the product of multiple factors, meaning a logarithmic function cannot improve accuracy in this case.

Future work will focus more on the issue of accuracy, but the current results may hold enough accuracy to use for preliminary results due to the large range of floating point numbers where some accuracy and the exponent are preserved.

6.3 Analysis

The Heterogeneous Naïve Bayes Classification ran with the PubMed and Clinical Trial datasets. Because this algorithm is unidirectional, two separate tests were done, one finding the relational ratings for PubMed articles mapped to Clinical Trials and the second test finding the relational ratings for the Clinical Trials mapped to the PubMed articles. These ratings were then sent to two separate output files that underwent analysis.

Firstly, the pairs of documents that had explicit relations before the algorithm should have high ratings. This held true for both output files. We can view C_n as the n^{th} document from Clinical Trials and M_n as the n^{th} document and $R_{m,c}$ as the relation between PubMed article m and Clinical Trial document c. $R_{m,c} = 1$ if an explicit relation exists between the two document pairs and $R_{m,c} = 0$ if no explicit relation exists between M_m and C_c. If $P(C_c|M_m)$ ends up with a high rating and $R_{m,c} = 0$, then one can assume that $P(C_c|M_z)$ also holds a high rating where $R_{z,c} = 1$ and that M_m and M_z most likely have similar contents in their documents. This should even apply to documents not in the training data as well.

A number of documents were looked at from PubMed where the ratings were high in relation to documents from Clinical Trials, but no explicit relation existed. Then, the contents of these PubMed documents with the PubMed documents that were explicitly related to the Clinical Trials at hand were manually investigated. In other words, we compared M_m such that $R_{m,c} = 0$ with M_z such that $R_{z,c} = 1$ where $P(C_c|M_m) \cong P(C_c|M_z)$. One such case involved $R_{458,414} = 1$ where $P(C_{414}|M_2)$ was highly related even though$R_{2,414} = 0$. When manually investigating the contents of M_{458} and M_2, one finds that M_{458} focuses on immunotherapy for breast cancer, while M_2 focuses on adjuvant chemotherapy in non-small-lung cancer. When taking in to account that this particular heterogeneous dataset consists of all cancer-related documents, then these two should be relatively different, which are, subjectively, undesired results. The results are similar when analyzing $P(M_m|C_c)$.

The poor results seems to stem from the lack of explicit relations in the original training data. With more explicit relations, we expect more desired results. Also, documents that seem to user likely words, such as “cancer”, more often than other documents, tend to add weight to the probabilities of the other documents as relations are calculated.

6.4 Future Additions

Due to the results of Heterogeneous Naïve Bayes Classification, it is apparent that alterations to the algorithm need to be considered in addition to the improvement of accuracy. Also, as is usual with learning algorithms, lack of training data leads to poor results. Future work would look for larger data sets with more explicit relations.

7. Discussions

From the results of limited trails, we only found the k=8 and rhoDivD=0.001can make dependent clustering work. And the result is not as good as we expected. The reason is probably because of the algorithm is sensitive to the data and parameters.

In addtiona, from the results above, we found several interesting findings: first, we found the two ASC plots are very similar, while the plots of percentage of biggest cluster are different. Second, for the clustering before the dependent clustering, using the entity as the feature is better for Clinical Trial dataset, and using each term as a feature is better for PubMed dataset. For the PC plots, the last dataset have different patterns from previous one. Third, in terms of percentage of the largest cluster, the using each entity as a feature outperforms using each term as a feature for clinical trial data, but for PubMed data is opposite.

In future, we need to be more focused on preprocessing the data, try to analyze terms and eliminated the irrelative ones. Also we should try different combination of parameters. And from the analyzing of the terms of each cluster, we think the cluster before dependent clustering failed to put these documents into right clusters, and that might be the reason the algorithm is not working well. So what we can try in future is to find a better way to classify the documents. For example, use supervised learning to classify all the document into different cancer types, or go back to

original abstract, and extract the cancer related terms and classify the documents based on that.

Furthermore, we wish to incorporate the suggested relations to investigate if the dependent clustering algorithm improves.

8. ACKNOWLEDGMENTS

This work is under the supervision of Dr. M. Shahriar Hossain, and supported in part by The University of Texas at El Paso.

9. REFERENCES

[1] ClinicalTrials.gov, A service of the U.S. National Institutes of Health. Retrieved February 9, 2015 from: https://clinicaltrials.gov/

[2] PubMed.gov, US National Library of Medicine, National Institutes of Health. Retrieved Feruary 9, 2015 from: http://www.ncbi.nlm.nih.gov/pubmed

[3] Hossain, M.S., Tadepalli, S., Watson, L.T., Davidson, I., Helm, R.F.& Ramakrishnan, N. (2010, July). Unifying dependent clustering and disparate clustering for non-homogeneous data. In proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining (pp. 593-602). ACM.

[4] Retrieved February 10, 2015 from Wikipedia, the free encyclopedia: http://en.wikipedia.org/wiki/Tf-idf

[5] http://alias-i.com/lingpipe/demos/tutorial/ne/read-me.html

[6] http://alias-i.com/lingpipe/web/models.html

[7] Cluster Analysis: Basic Concepts and Algorithms. Retrieved February 11, 2015 from: http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf

[8] Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1), 37-52.