Liam Pronovost
COMP 4334
Midterm Project
10/12/2022

Problem:

The goal of this project was to uncover underlying factors of medical conditions by applying machine learning on a provided data set. The data set contained both txt and ann files of doctors notes (303 of each). Each text file contained a detailed report of the patient's visit, history, medication, and condition. While the text files were rather variable, there were a number of similar features throughout the notes, three of them being "chief complaint", "history of present illness" (HPI), and "discharge diagnosis." The ann files primarily contained information on the medication, categorizing information into "drug", "dosage", "frequency, "route", and "reason."

Solution:

Plan:

My overall solution to this problem is described below:

1. Data Extraction, Formatting, and Cleaning
2. Data Exploration and Examination
3. Model Fitting (LDA Topic Modeling) and Selection (Word Intrusion and Coherence)
4. Model Tuning

Data Extraction and Formatting:

My first step in this project was to extract medical conditions from each doctor's note. There are a number of conditions and complaints described in a doctor's note, but the primary reason for the visit is often found in the "chief complaint" section. Initially, my plan was to use this section to extract the overall diagnosis of the note. However, from my previous experience scribing doctors notes, I know the chief complaint is typically provided by the patient. More often than not, a chief complaint is a symptom, such as nausea or abdominal pain, rather than a medical diagnosis. Consequently, I decided to use the "discharge diagnosis", which is provided by the physician, as the source of the overall diagnosis, and compare the two methods. To find medical conditions in these sections, I used SciSpacy's NER. There are a number of useful NER libraries within SciSpacy. The "en_ner_bc5cdr" library contained a "DISEASE" entity, which made it the most fitting library for this problem. Using both the chief complaint and discharge diagnosis, the resulting conditions were mapped to a list of files pertaining to them, and the number occurrences of each disease was counted (see code comments).

Next, a bag of words (BoW) for each note needed to be created. Originally, I tokenized each note, removed stop words, and modeled its respective corpus. The resulting topics contained no diagnoses or symptoms and mainly contained words like "patient", "mg", "daily", "po", etc. After adding these words to my list of stop words and running the model again, there were still no diagnoses or symptoms. Rather than continuing to update a large list of stop words, I once again used SciSpacy's NER to only consider diseases in my BoW (a bag of diseases). I employed the NER twice, first on the entire note, second on the HPI separately. I wanted to examine which bag of diseases (BoD) would generate a better model. The BoD of each note was put into a data frame with its respective text file.

Data Cleaning:

I performed my data cleaning while extracting and formatting the data. Other than removing special characters and numbers, the cleaning mainly involved fixing some common "diseases" the NER categorized. First, I removed common prefixes to diagnoses (e.g changing "chronic kidney failure" → "kidney failure"). Second, I simplified diseases with extra language (e.g changing "diabete mellitus" → "diabetes"). This way my model wasn't fitted to overly specific or extra wordy diagnoses. Finally, I changed common occurrences of abbreviations, spelling mistakes, or terminology that related to a disease (e.g "htn" → "hypertension", "hypertensive" → 'hypertension"). All of this cleaning allowed for my BoD to contain general, consistent, common diseases. Much of my cleaning was manual and involved thorough examination and re-running of my model.

Data Exploration and Examination:

The first step in exploration involved clustering. Using TF-IDF and PCA, the documents were converted into a clusterable format. Using an elbow plot, the notes were grouped into eight clusters. After clustering, it was clear that data was fairly closely grouped, with two groups of outliers. Outside of clustering, the data exploration primarily involved examining the difference between using the "chief complaint" and "discharge diagnosis." At first glance, "discharge diagnosis" seemed to immediately perform better. The counts of diagnoses were much higher when using the discharge diagnosis section. But why does this matter? When I examined the diagnoses using each method, the more frequent diagnoses were recognizable medical conditions. However, as the count lowered, the diagnoses became less recognizable and less generally applicable. Furthermore, a lower frequency meant that diagnosis applied to less data, which posed a risk of underfitting. The discharge diagnosis method provided higher frequencies of diagnoses, consequently resulting in a larger dataset (see figures 1.1 and 1.2). Because of this, I chose to move forward with the discharge diagnosis method. Using 125 of the most frequent medical conditions, roughly ~220 notes would be included in our dataset.

Model Fitting and Selection:

I decided to fit my model using Latent Dirichlet Allocation (LDA) Topic Modeling (gensim's LdaMulticore model). This model employs a probabilistic approach which allowed me to extract the best features based on statistical measurements. Two different data sets were provided to the model. One dataset was composed of BoDs from the HPI only. The other data set contained BoD's from the entire note.

For each data set, all the BoD's pertaining to a given medical diagnosis (extracted from the discharge diagnosis) were used to train the model. Once the model was fit, the features (diseases) were extracted, and their probabilities were totaled. If the probability of a condition appearing in a corpus was 1.5 standard deviations above the average probability within that corpus, the condition was considered to be an underlying factor of the given medical condition. This was done for all 125 diagnoses.

Comparing the results of the different datasets was rather difficult, but I believe the model fitted only to the HPI performed better. In order to compare the models, I manually looked for Word Intrusion. Technically, each model used a different dataset (HPI corpus and entire note corpus). Therefore, coherence was not applicable (that was used in model tuning). While both models often had similar underlying conditions, models using the entire note had more frequent word intrusion. In the model using the entire note, hypertension and hypotension were very frequent and often appeared together, which caused some concern seeing that they are opposites. Hypertension and hypotension were much less frequent in the HPI model, and rarely appeared together.

Thinking through the results, it made sense that the model using the entire note resulted in more word intrusion. The entire note includes diseases the patient is medicated for, negative conditions (e.g "patient was *not* hypotensive") and conditions listed in the review of systems (ROS). This most likely added conditions that did not necessarily pertain to the discharge diagnosis. Due to this, I was confident with selecting the HPI model, and decided to tune this model. See figure 2.1 for resulting underlying condition counts.

Model Tuning

Two features were hypertuned in this model: the alpha and the decay. Decay weights what percentage of the previous lambda is forgotten between documents while alpha affects the weight of the number of topics. The decay values used were 0.6, 0.8, and 1.0 and the alpha values used were 0.1, 0.5, and 1. This totaled to a combination of 9 different models fitted to the data. When run on all 125 diagnoses, the combination of models took about 2 hours to run (1,125 LDA models were fitted). For the sake of run time and analysis, the model was hypertuned with only the top 20 diagnoses. The average coherence was calculated and plotted (see figure 3.1). Between the tuned models, it seemed that an alpha of 0.1 and a decay of 1.0 achieved the best average coherence and slightly outperformed the base model.

Analysis:

Overall, I believe my solution worked fairly well, particularly on the more common diagnoses. These top diagnoses had clear and understandable underlying conditions, with little to no word intrusion. Using the Discharge Diagnosis improved the size and clarity of my dataset, and using the HPI led to more accurate results. While tuning the model improved the average coherence, the resulting underlying conditions did not seem to be significantly different.

The model could be improved in a number of ways. Unfortunately, due to the time constraint, these were unable to be attempted. I was unable to see how the model performed on data it had not yet seen (seeing the probability a given HPI relates to a topic/medical condition). Another possible improvement would involve extraction of the negative conditions from the ROS to then further evaluate the model performance (see if those negative conditions are still considered underlying factors).

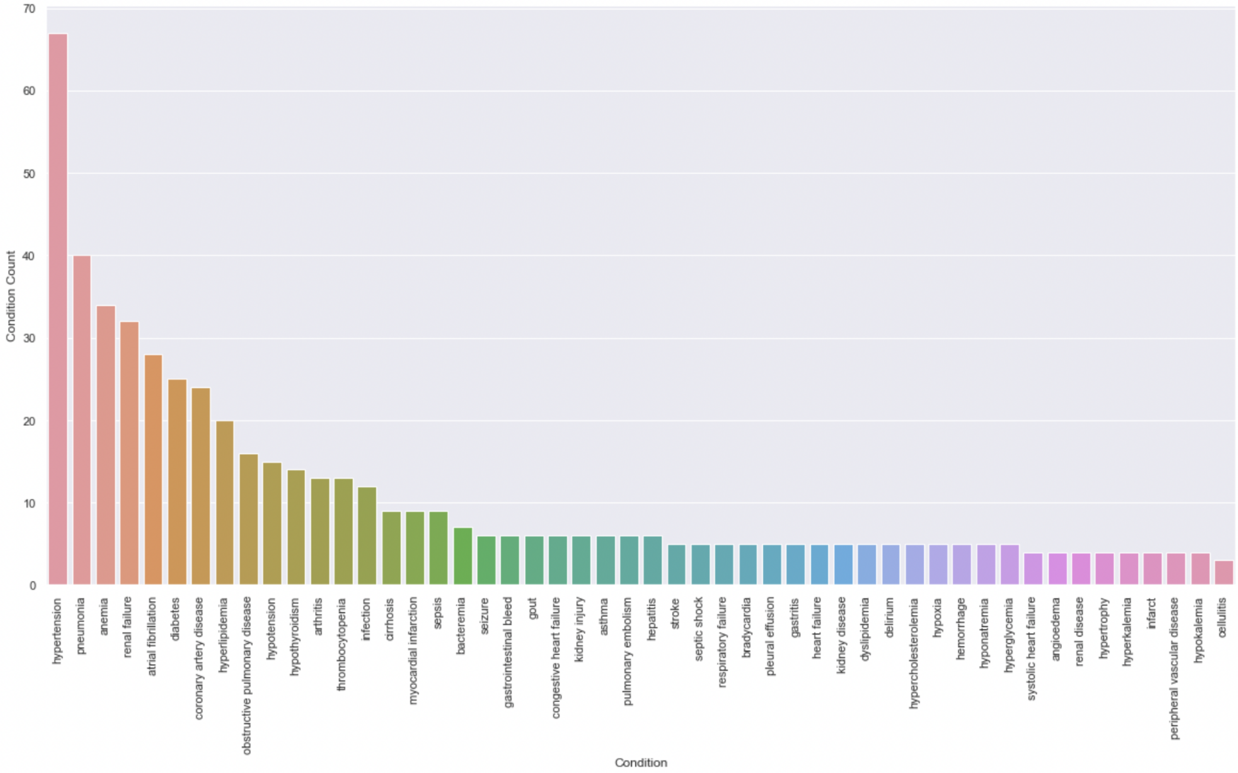
To my reviewers:

Thank you for your time and consideration. This project has been a great learning opportunity for me and I honestly enjoyed working on it. I loved getting to utilize my healthcare experience in a new way, and hope it provided a creative and unique approach to the problem. I look forward to getting the opportunity to discuss my work.

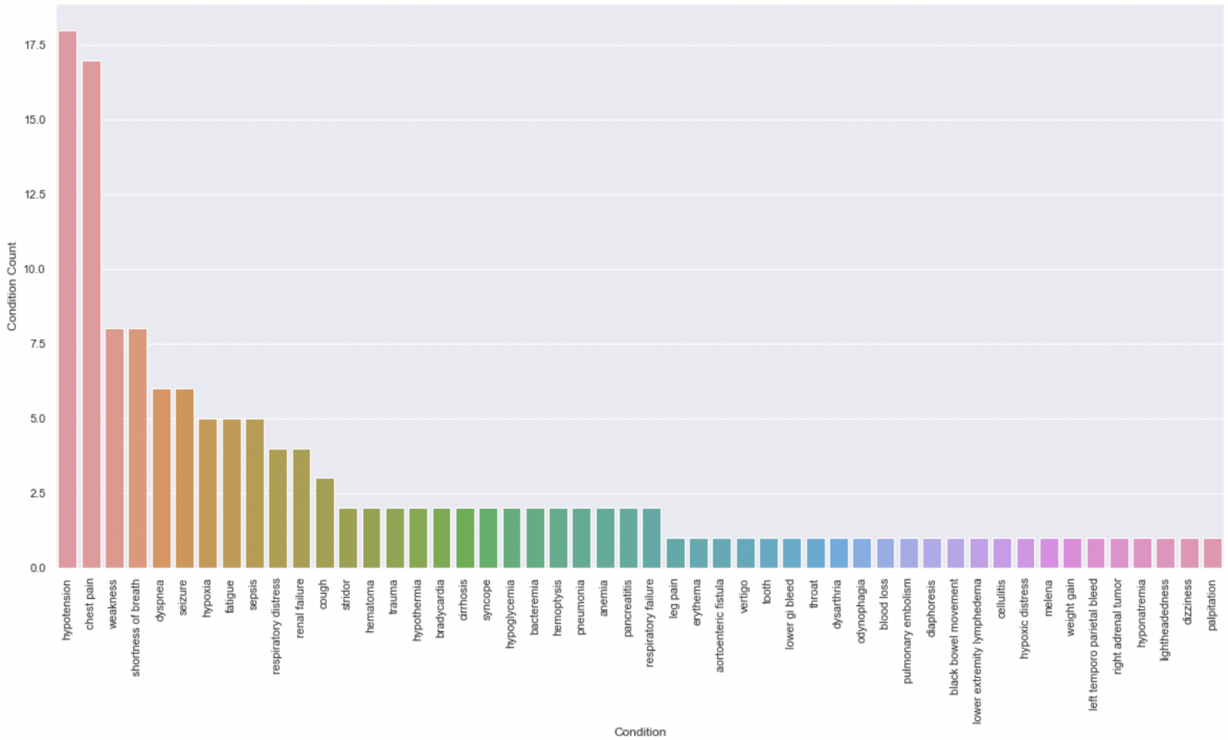
Figures:

1.1 Bar graph of Discharge Diagnosis condition counts and Chief Complaint condition counts

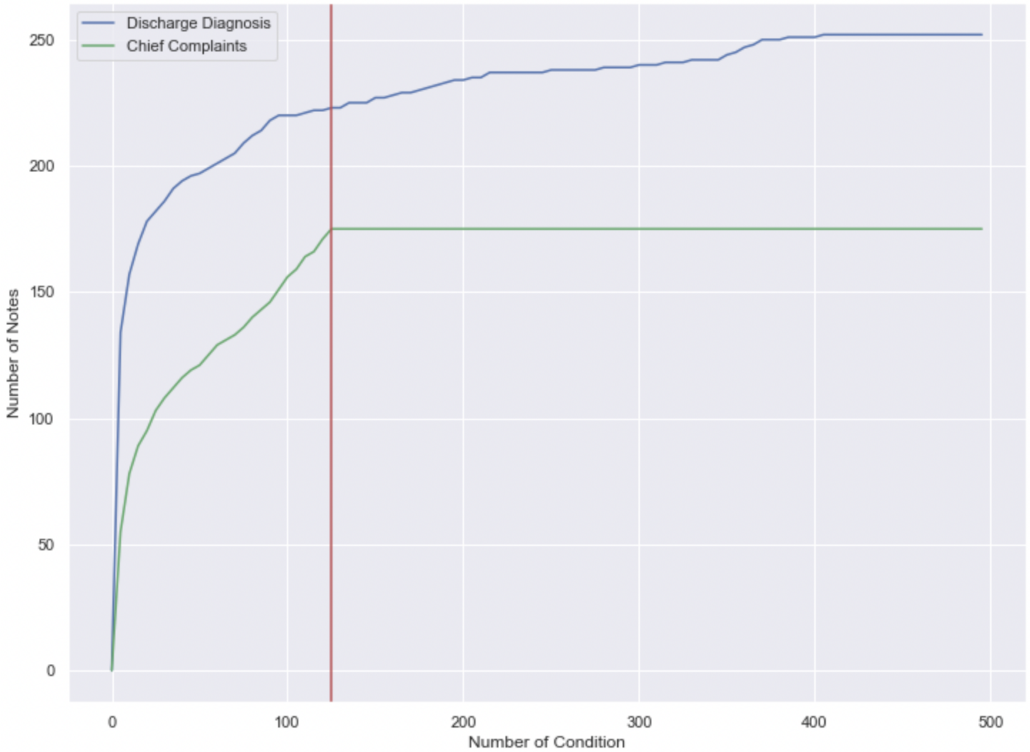
Discharge Diagnosis



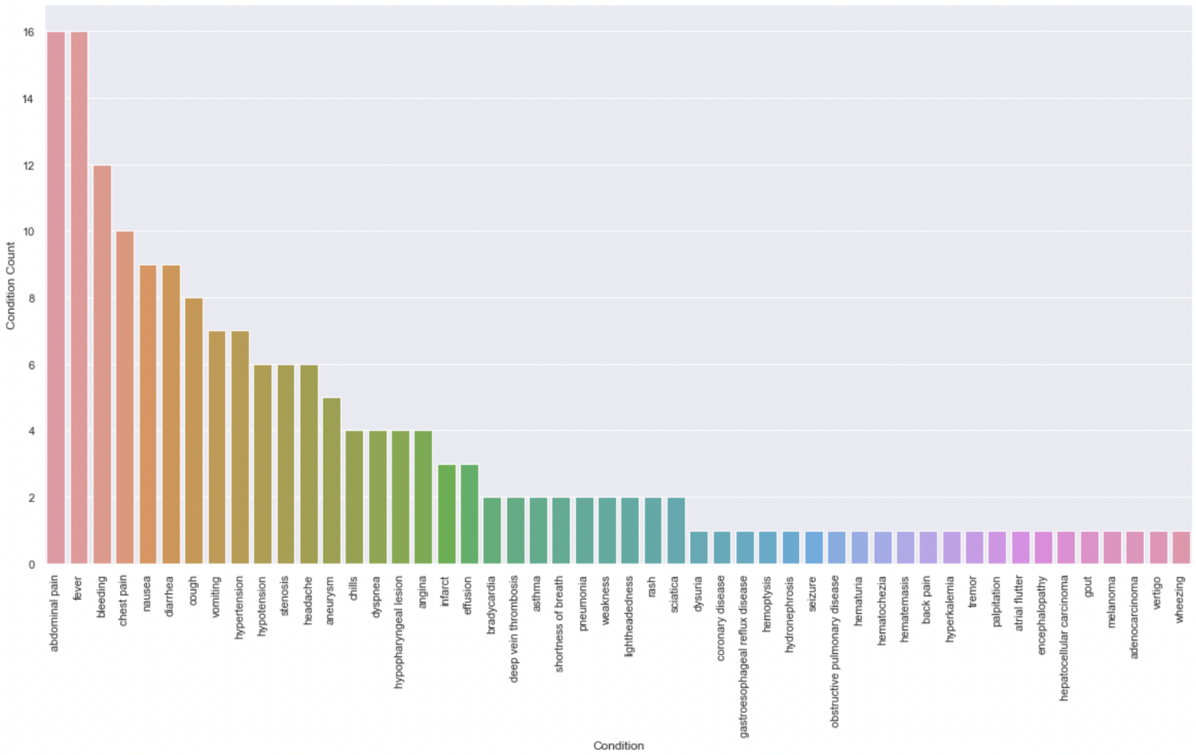
Chief Complaint



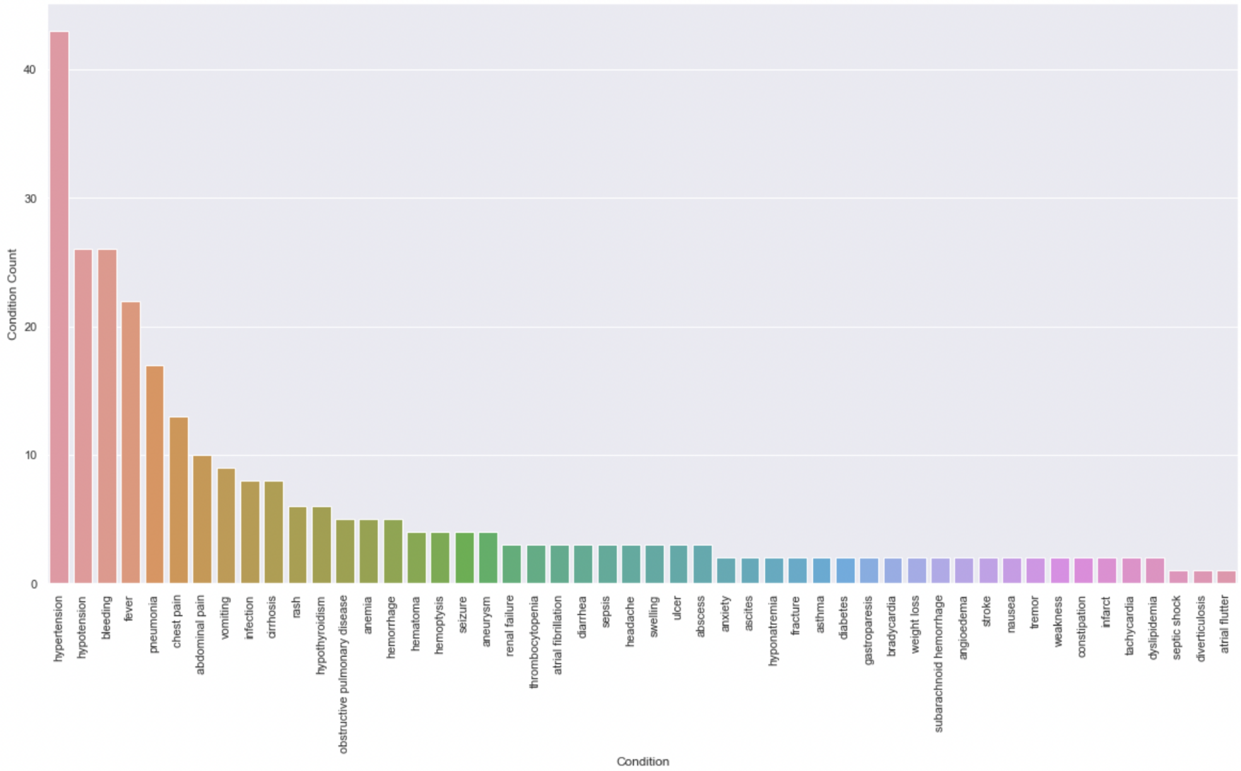
1.2 Graph of Number of Conditions vs Number of Notes included in the data set for each method. Red line was the chosen number of conditions (125).



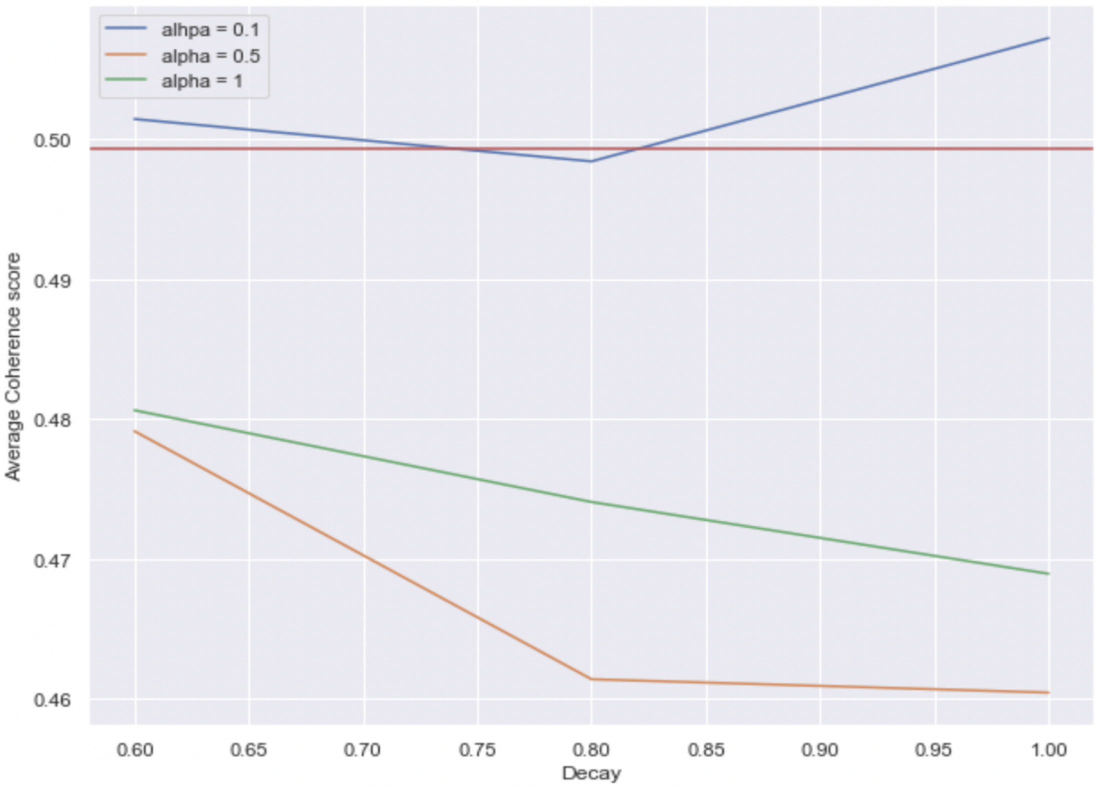
2.2 Bar Plots of underlying condition counts.
HPI only bag of diseases:



Entire Note bag of diseases:



3.1 Plot of Average coherence vs decay for alphas of 0.1, 0.5 and 1 (red line is base model).


alhpa = 0.1
alpha = 0.5
alpha = 1
0.50
0.49
0.48
0.47
0.46
Average Coherence score
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
1.00
Decay