Leveraging Deep Semantic Embeddings and Clustering of Patient Portal Messages to Identify Appointment Scheduling Barriers

Jasmine Bilir, Julia Kadie, Tran Le
Stanford University Department of Computer Science
jbilir@stanford.edu, jkadie@stanford.edu, tranle@stanford.edu

December 8, 2022

Abstract

At Stanford Medicine, patient portals have been widely used by healthcare professionals and patients to enable timely communications via secure messaging for various topics, including appointment scheduling. The large volume of portal messages offers an invaluable opportunity for studying appointment-related issues reported by patients. In this work, we leveraged cutting-edge deep learning techniques and deep semantics for identifying appointment issues mentioned in patient portal messages. Our results successfully identified referrals, timing, and travel as three common issues, and quantitative measures showed reasonable homogeneity and tight grouping within issue clusters. The successful creation of annotated corpus and identification of three appointment issues showed the feasibility of this strategy. The developed annotated corpus could aid in developing an artificial intelligence tool to automatically identify appointment issues from millions of patient portal messages. The identified appointment trends can be leveraged as guidance to reduce friction in the patient experience.

Keywords: Patient Portals Messages; Appointment Issues for Patients.

1 Introduction

Within medicine, secure messages are a critical tool for communication and collaboration within healthcare teams and with patients. At Stanford Health Care, the MyHealth application is used to communicate between the patient and care team. Through our interviews with clinical informatics fellow and internal medicine physician Dr. Julian Genkins at Stanford Health Care, we found that many messages relate to appointment scheduling in some way. Thus, our goal is to identify trends in why and how people encounter issues with their healthcare appointments. The input to our algorithm is unstructured patient message data. It includes messages to and from patients. See section 3 (Dataset and Features) for more details. Our output will be identification of a list of appointment issues related to scheduling, rescheduling, and canceling. We obtained this dataset by working with Professor Jonathan Chen's Health Rex Lab. This dataset is unique to Stanford Medicine patients. We completed training including the HIPAA for biomedical research to have access to this dataset. As such, deep learning research has not yet been conducted on this dataset. Moreover, our literature review showed a gap in research related to using AI-powered techniques for identifying appointment barriers, reflecting the nuance of our idea.

2 Literature Review

2.1 Understanding patient experience from online medium [7].

In the dissertation paper "Understanding patient experience from online medium", Cho et. al. discuss various approaches to semantic word encodings specifically to be used in understanding patient generated texts (i.e. messages sent to medical facilities like those in our dataset, or other online texts such as WebMD etc). In this analysis of outstanding research, it is emphasized that patient-generated texts do not fair well under standard bag of words encoding. Because of the proximity of medically related topics, gaps in patient understanding, as well as patient likelihood to bring up multiple ideas in one message, more targeted word encodings are necessary. Since we will be using SBERT in combination with modifications and other semantic encoding techniques, we reviewed the sections of this article to fine-tune our model after a baseline was achieved. In addition, following this paper's findings, we attempted to create our own SBERT model from scratch to see if it would perform better with medical data.

2.2 Breast cancer symptom clusters derived from social media and research study data using improved k-medoid clustering [8].

In this research study, Ping et. al. reviewed a dataset containing patient reports in research and online and used k-medoid clustering in order to better understand how symptoms related to one another. The study used a total of 50426 patient written texts, and the results were evaluated using average silhouette width. We used this paper to compare each phase of our model and to see what information might be useful in structuring our model. However, our methodology differs significantly due to our use of hierarchical and fast clustering, a decision made since we believe that these deep learning clustering methods would better fit our task.

2.3 Symptom clustering in advanced cancer [9].

In this retrospective study, researchers used 1,000 consecutive patients and their data to cluster symptoms based on presence and severity. Agglomerative hierarchical clustering was used in order to build these clusters and later to more closely link clusters for a more reasonable statistical viewing. Overall, the study found that the overall correlation was ≥ 0.68 for all final clusters. This research paper showed how hierarchical clustering could be used with medical data. We leveraged the knowledge for applications of this paper's clustering model for our own research.

2.4 Identification of transportation barriers in patient portal messages via deep semantic embeddings and clustering [10].

This excellent research paper was a large inspiration for our project. Huang et. al. leveraged deep semantic embeddings and clustering methods to identity transportation issues from patient portal messages. The researchers used similar methods to what we performed in this research. However, we also tried other methods like training a Siamese BERT (S-BERT) model from scratch and using Global Vectors for Word Representation (GloVe) to see if we can achieve better results. The dataset and focus area in transportation were also different from ours.

2.5 Inpatient portal clusters: identifying user groups based on portal features [11].

Fareed et. al. leveraged the hierarchical clustering algorithm to improve understanding of who uses the patient message portals and how. The results showed that common usage includes reviewing schedules, results, tutorials, and ordering food. Although the portal is not the same as the Stanford Medicine portal, the paper provided valuable insights on the feasibility of using deep learning clustering methods to understand patient portal messages.

3 Dataset and Features

Our dataset includes the following items: 1) the message that the patient or medical provider sent (We have a total of 391,385 messages in our dataset), 2) a unique anonymous ID for each message.

We obtained the dataset from the Stanford Health Rex Labs. The dataset contained raw messages from Stanford patient portal messages that had not been preprocessed. We considered messages from patients of all demographics in our model. We conducted a preliminary analysis examining how many messages out of 391,385 contain the following two keywords corresponding to appointments: "appointment", "appointments". This analysis showed promising amounts of data related to appointment scheduling in our messages dataset, with 24% of total messages containing the words "appointment" and "appointments", making up 95,197 messages.

For our data set, we cleaned the data by removing stop words, tokenizing the sentences, and lemmatizing the words using Natural Language Toolkit (NLTK). This ensured that only the most important information of the messages is included for training. The complete code for our data cleaning method can be seen here.

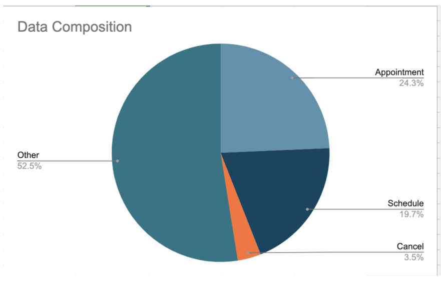


Figure 1: Pie Chart of Dataset Composition

4 Baseline Methods

After cleaning our dataset, we proceeded to generate deep semantic embeddings of the message using Siamese BERT. We used the "all-MiniLM-L6-v2" model from HuggingFace [4].

After getting deep semantic embeddings on all our messages, we performed clustering to find patterns in the messages. We chose k-means as our baseline clustering algorithm with a hard coded number of 5 clusters. Then, we sampled 10 sentences from each cluster. We manually examined the sentences to determine the trends in each of the five cluster to determine the issues in appointments, including: (1) patients are uncertain on when is the best time to schedule an appointment, (2) patients experience a lack of information confirmation before the appointment, (3) patients have questions about tests and prescriptions before the appointment, (4) patients need more instructions before and after the appointment on and (5) patients are uncertain about which doctor they should schedule an appointment with.

These initial results showed that there is feasibility and novelty in the research topic for us to continue the project. However, the baseline method did not produce significant results because the pre-trained model was not tuned to our data, not all the messages in the data had appointment issues, and K-means is a naive clustering technique. Specifically, we had concerns that these clusters were neither homogeneous or tight meaning that there was a great deal of overlap and variability in how clusters were assigned.

5 Methods

5.1 Methods Overview

Figure 1 shows our overall workflow to identify appointment issues mentioned in patient portal messages. First, we wrote SQL queries to collect the dataset as mentioned in section 3. The query also identified and merged sentences belonging to the same message together in one data point. Once we have the messages, we tokenized them into sentences. We then used the simple keyword searching method to search for messages containing "appointment". Once we found these sentences, we manually annotated 50 of these sentences on whether the message mentions appointment-related issues. This served as our starting seed dataset. After converting the sentences to deep semantic embeddings, we then expanded the seed dataset by searching semantically similar sentences in an iterative manner. We manually checked the accuracy by checking random outputs. Once we had a large enough seed dataset with more than 200

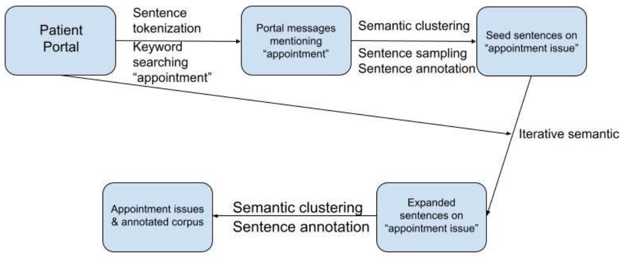

Patient Portal
Sentence tokenization
Keyword searching "appointment"
Portal messages mentioning "appointment"
Semantic clustering
Sentence sampling
Sentence annotation
Seed sentences on "appointment issue"
Iterative semantic
Expanded sentences on "appointment issue"
Semantic clustering
Sentence annotation
Appointment issues & annotated corpus

Figure 2: Workflow to identify appointment issues in patient portal messages

messages containing appointment issues, we applied clustering analysis to learn about specific trends of appointment issues as mentioned by patients in their portal messages.

5.2 Data Cleaning

To tokenize each message, we used the NLTK sentence chunk package [12]. To find messages that contains "appointment", we wrote a linear search Python script. To generate deep semantic embeddings of the messages that contains "appointment", we experimented with using a pretrained SBERT model, self-trained SBERT model, and pretrained GloVe model.

5.3 Deep Semantic Embeddings and Clustering

Next, we clustered the larger data set and appointment issue specific data set into groups of similar semantic embedded sentences. We explored using three clustering methods: hierarchical agglomerative clustering [2], Density-Based Clustering Based on Hierarchical Density Estimates (hdbscan) [3], and SBERT's native fast clustering [4]. We decided to move forward with fast clustering because hierarchical agglomerative clustering and fast clustering are quite slow on larger datasets. In order to use it and extract meaningful results, we would have had to decrease the size of our original dataset to only a few thousand sentences. Ultimately, we found fast clustering to have the best performance so we moved forward with this technique. Fast clustering allowed us to configure the threshold of cosine-similarity for which we consider two sentences as similar and enabled us to specify the minimal size for a local community. By tuning these two hyperparameters, we were able to optimize our quantitative and qualitative metrics as described below and ultimately get large coarse-grained clusters of appointment issues.

To further improve our seed dataset amd develop a fully annotated corpus, we used an iterative process using cosine similarity to identify the most relevant messages. First, we converted all the messages to vector embeddings using 50D pre-trained GloVe model. We then calculated the pairwise cosine similarity between our seed dataset and every sample in the clean dataset. We identified the top 10 closest matches and manually annotated whether appointment issues were mentioned. Any sentences that were verified to contain appointment issues were added to the seed dataset and the process was repeated for a total of 25 iterations to gather over 200 confirmed samples. We then used these verified samples to make up our completed corpus to be used for our SBERT model and clustering operations.

We used a GloVe model for this stage of processing for multiple reasons. GloVe is a non-contextual global vectorization model, and we believed this might be a useful way of grouping the data in ways that might not implicitly be contained in the medical texts we were using. Likewise, we also used human generated text, much of the formatting differed so we hoped that using a context-free embedding model might yield more accurate pairwise similarities given the global context-free nature of GloVe. Lastly, we viewed this as an opportunity for comparison between the original dataset and this annotated one as well as to attempt a novel approach relative to Huang et. al.

5.4 Topic Analysis

After clustering messages with the methods in **5.3**, we conducted topic analysis using Top2Vec to identify common topics in each cluster. Top2Vec leverages joint document and word semantic embedding to find topic vectors [13]. The algorithm assumes that semantically similar documents share common topics and uses embeddings to find dense clusters of documents, and identify which words attracted those documents together. To supplement Top2Vec, we also used the BERTopic algorithm to generate the most relevant words per cluster. BERTopic is a topic modeling technique that leverages transformers and c-TF-IDF to create dense clusters allowing for easily interpretable topics while keeping important words in the topic descriptions [17]. This methodology helped us focus our search when manually annotating clusters to seek further clarification on the meanings of topics generated by top2vec.

6 Experiments, Results, and Discussion

6.1 Experiment: Training an SBERT model from scratch

In addition to using the pre-trained SBERT model to create deep semantic embedding of the messages, we also attempted to train our own SBERT model on the data that we had. We used the following process to create the model:

1. We created the network architectures from scratch by defining the individual layers. First, we limited the layer to a maximal sequence length of 256 so that all embeddings will have the same features. Then, we created a mean pooling layer because we want to map a variable length input text to a fixed sized dense vector. With mean-pooling, we averaged all contextualized word embeddings BERT gives us. Then, we create a new SentenceTransformer model with the Tanh activation function, which performs a down-project to 256 dimensions. We chose the Tanh function because it outperforms the sigmoid function by centering the weights at 0.

2. Next, we trained the network on self-labeled data. We randomly selected 26 messages. Then, we created random pairs of sentences. Then, for each pair of sentence, we gave a label from 0 to 1. 0 means that the two sentences are very different, and 1 means that the sentences are very similar.

3. For our loss function, we use the cosine similarity loss function. We chose this function because it is well-suited for the task of recognizing the similarity between sentences.

4. For our evaluator, we used the embedding similarity evaluator [1] which measures the similarity of the model's embeddings to the embeddings of test data that we pass in.

5. We trained and used the self-trained sentence transformer model on our full dataset. Then, we performed the same K-means clustering with 5 clusters as we originally did with the pre-trained SBERT model.

After training the model, we received a EmbeddingSimilarityEvaluator of 0.346 which means that the model's embeddings had a 34.6% similarity to the evaluator's embedding test data set. This shows that the network trained from scratch was not very accurate. We then sampled 10 sentences from each cluster to examine whether sentences in each cluster are related and distinct from other clusters. After examining the clusters, we realized that the model was not clustering the sentences into separate groups with identifiable trends as compared to using the pre-trained SBERT model. For example, in the first cluster, two randomly sampled messages state different problems. The first message states a problem about a user error of the online patient messaging portal and trying to move an appointment date. The second message states a problem about not receiving enough instructions before the appointment. By quantitatively analyzing the data, along with the similarity evaluator, we clearly saw that the self-trained network was not accurate in embedding the sentences, given that we have used the same clustering method in both models.

We hypothesize that the biggest bottleneck with the poor performance of our self-trained model was the quality and quantity of the training data set. If we had more time and resources, we would improve this model by self-labeling more pairs of sentences so that the model has more data to learn from. However, with our current constraints, using the pre-trained SBERT model with millions of sentences in the data set was a better approach.

6.2 Results and evaluation

For the identification of appointment issues, we leveraged fast clustering, (as mentioned in section **4.3**) to group together semantic embeddings from the pre-trained SBERT model.

6.2.1 Quantitative evaluation

To analyze our clusters and tune our hyper-parameters, we used a combination of Silhouette Scoring [15] and Within-Sum-Square-Errors (WSS)[14] which indicated to us how closely our clusters are correlated within themselves.

In tuning our model, specifically looking at parameters like number of clusters and minimum community size, our goal was to minimize WSS [16] score, as this indicates that our groups were both tight and homogeneous. Likewise, we aimed to achieve a silhouette score (which ranges between -1 and 1) that was as positive as possible, as a negative silhouette score indicates that data points were incorrectly assigned. We noticed that regardless of our parameter tuning we always reached relatively small scores in the range of 0.01 to 0.1. However, after reading relevant works and examining our data, we attributed this to the similarity of messages in our data set and even more-so in our cosine similarity data set. Therefore, we recognize that silhouette scoring also provides information about the construction of our dataset, not just the performance of our model. We also explored the inclusion of other sum-squared-error metrics including Between-Cluster-Sum-Squares, Total-Cluster-Sum-Squares, R^2 metrics, but in combination with our qualitative metrics, we found these to be less instrumental in understanding our model's performance.

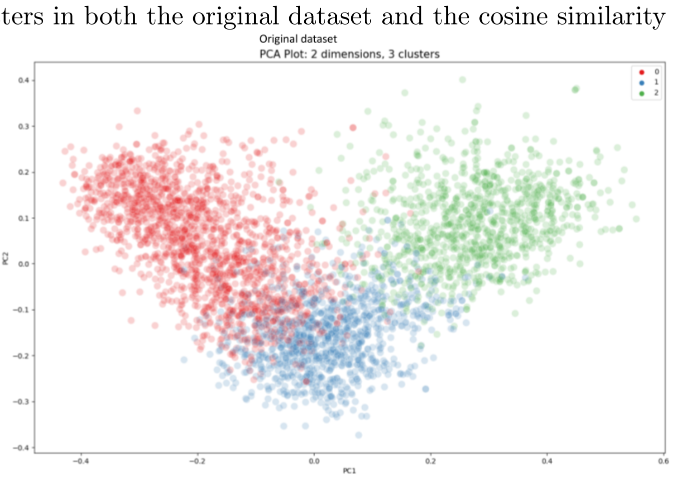
Data Set	Min Community Size	Num clusters	Silhouette Score	WSS
Full	1000	3	0.110420324	87.7018597179
Cosine-Sim	5	6	0.07496815	1.450808727777

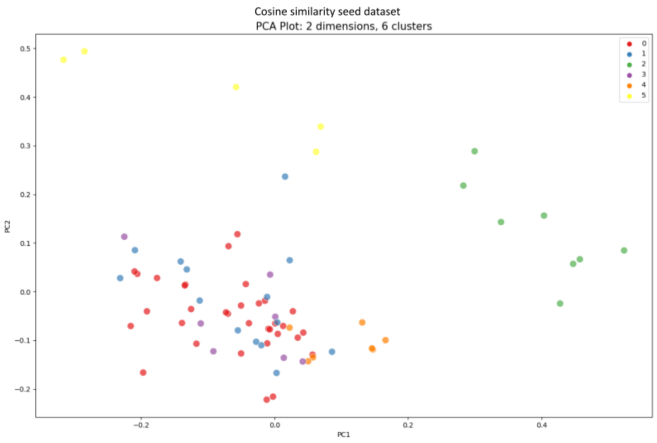
The results above feature our best quantitative performance after trying many combinations of hyper-parameter tuning. We believe the higher performance of the full-data set was due to the sizing issues (see conclusion for further details). For more examples of our quantitative metric data, see the appendix.

6.2.2 Qualitative evaluation

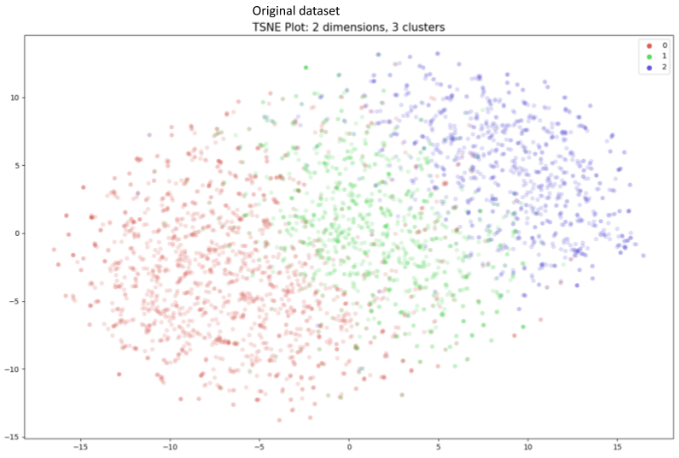
Since our semantic embeddings for our messages were of dimension 284, in order to visualize the quality of our clusters, we leveraged two dimensionality reduction techniques: Principal Component Analysis (PCA) [5] and t-Stochastic Neighbor Embedding (t-SNE)[6].

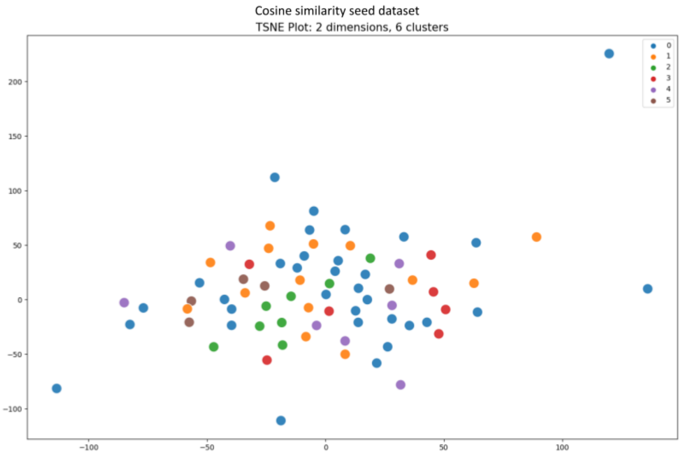
PCA: PCA is the most commonly used tool for dimensionality reduction, and is used in many fields for understanding multidimensional data. We used PCA to reduce our embeddings' dimensions from 284 to 3. We produced the following graphics, color coding our PCA-reduced datapoints according to the clusters as determined by the fast clustering algorithm mentioned in subsection 4.3. The PCA plots made confirm the quality of our clusters in both the original dataset and the cosine similarity dataset.





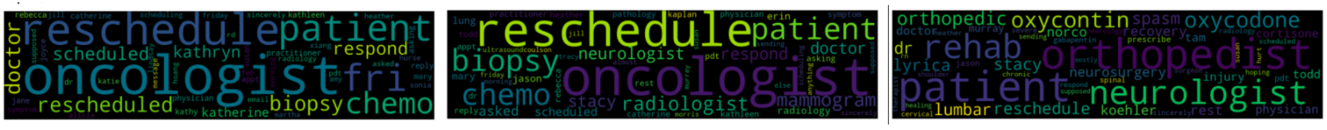
t-SNE t-SNE is another commonly-used dimensionality-reduction technique. It is a variation of Stochastic Neighbor Embedding that reduces the tendency to crowd points together in the center of the map. We used this technique to visualize our clusters again in 2d space. These t-SNE plots verified the efficacy of our clustering method on the original dataset. There are less definitive trends shown in the cosine similarity dataset's t-SNE plot, which is likely due to the small size of the dataset. We remain confident in our clustering because the PCA technique showed definitive groupings amongst clusters.





6.2.3 Identification of appointment issues with topic modeling

Figure 3: Cluster 1, 2, and 3



After sanity-checking our clustering with the dimensionality-reduction techniques mentioned in **5.2.1**, topic modeling was used to identify the common themes in each of the fine-tuned clusters. In the large dataset, there were 3 clusters identified. Using topic modeling, we made wordclouds that help us identify the topics that unify the clusters. Cluster 1 and Cluster 2 were similar in that their major issues of concern were regarding rescheduling cancer-related appointments. Cluster 3 primarily concerned prescriptions and specialists, likely concerning the need to go to a specialist to receive important prescriptions.

For the cosine-similarity-generated dataset, we used top2vec alongside a manual calculation of relevant topics to identify the topics most important in our 6 clusters. Due to the small size of the dataset, we used top2vec results to guide our manual interpretation of themes in each cluster. Out of the 6 clusters, 3 deterministically had topics. The themes found were: referral issues, travel issues, and timing issues. We present anonymized examples of messages from each category below:
Referral issues "Hi [name], Hope this email finds you well. I just called the sleep Clinic and they won't let me make an appointment without an updated referral from Dr.[name]. Could you please send them one? Thanks, [name]"
Timing issues "We currently have an appointment scheduled the week after that which includes the test needed and an appointment with the doctor for assessment. We are wondering if just waiting the additional week is the better choice now? It looks to me like the doctor is only in clinic on [Day] and must be out the first week of [month]. [Name] and I are out of town on [Day]."
Travel issues: "I do not feel it's in my best interest to drive all the way from [city] to Stanford, get a low blood count report there and not get an injection.... Please let me know what you want to do with the information above. If I will continue to have the appointment on the 31st, I will need to know right away so my husband can request the time off work."

7 Conclusion/Future Work

The results of our research revealed that some of the most substantial barriers to patients experiencing a difficulty before, during, and after their appointments are travel, timing, and referral related issues. These results could be used to improve the MyHealth patient portal app and used to improve the healthcare processes so that appointment processes are more streamlined for patients. We will be working with physicians to further research and implement our findings.

Our evaluations, both quantitative and qualitative, indicated that we achieved better results on the full dataset as opposed to the cosine similarity seed dataset. This was unexpected as the process of developing the manually annotated dataset was intended to improve our results as it did in our referenced papers. We believe that this is primarily due to a sizing issue. We think that regardless of our efforts to highlight appointment issues, the large size difference (full-data set with 391,385 samples and cosine similarity data set with 285 examples) accounted for the loss in performance. With a greater number of iterations and data samples, we think the annotated data set would have achieved similar if not better results. However, in both datasets, our quantitative and qualitative measures showed significant improvement from our K-Means baseline. Not only did we uncover more specific trends via word clouds and Top2Vec, but we developed clusters that showed quantitative indications of being tight and homogeneous.

If we had more time and resources, we think the following changes might have improved our results based on our understanding of our present results: increase size of cosine similarity seed data set by performing more iterations, further pre-processing to remove names or individualized information (these impacted clusters heavily in smaller datasets), ensure that every patient message is independent of other patient messages (messages that were replies/in the same conversation were likely to be clustered together).

8 Contributions

Each team member equally contributed to the research and ensured the decisions were unbiased.

Tran worked on querying, preprocessing, and performing initial data analysis to brainstorm research areas and ensure the project proposal in the appointment issues space was feasible. She also trained the SBERT model from scratch with multiple different parameter experiments and ran K-means clustering on the self-trained SBERT model to see if it would be a more accurate way to embed and cluster data.

Jasmine worked on developing the initial k-means algorithm and experimenting with pre-trained SBERT models in order to verify that the project and the question being explored were capable of delivering worthwhile results for the proposal and milestone 1. She also wrote the code for and conducted the cosine similarity expansion for the fully annotated data set. Jasmine also focused on writing code for and analyzing the quantitative metrics.

Julia worked on and wrote code for the 3 clustering techniques that we considered (hierarchical agglomerative clustering, hdbscan, and fast clustering. After selecting fast clustering for its speed benefits, she fine-tuned the fast-clustering model. In the end, Julia tried 25 parameter combinations for this model. She also conducted and wrote code for the two dimensionality-reduction techniques, PCA and t-SNE, and created plots for the reduced-dimension data. Julia conducted and wrote code for the topic analysis component of the project, implementing manual topic analysis with the BERTopic algorithm to guide manual annotation of clusters' topics and wrote code leveraging top2vec to generate topics and visualizations for each cluster.

References

[1] UKPLab. “UKPLAB/Sentence-Transformers: Multilingual Sentence amp; Image Embeddings with Bert.” GitHub, https://github.com/UKPLab/sentence-transformers.

[2] Jovana, et al. “Agglomerative Hierarchical Clustering.” Datanovia, 20 Oct. 2018, https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering/.

[3] Campello, R.J.G.B., Moulavi, D., Sander, J. (2013). Density-Based Clustering Based on Hierarchical Density Estimates. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2013. Lecture Notes in Computer Science(), vol 7819. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-37456-2_14

[4] “Clustering Methods with SBERT.” Clustering - Sentence-Transformers Documentation, https://www.sbert.net/examples/applications/clustering/README.html.

[5]Timmerman, M.E. Principal Component Analysis. J. Am. Stat. Assoc. 2003, 98, 464.

[6] Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. J. Mach. Learn. Res. 2008, 9, 2579–2625.

[7] Cho, Hyun Duk. Understanding patient experience from online medium. Diss. University of Illinois at Urbana-Champaign, 2019.

[8] Ping, Qing, et al. "Breast cancer symptom clusters derived from social media and research study data using improved *k*-medoid clustering." IEEE transactions on computational social systems 3.2 (2016): 63-74.

[9]Walsh, Declan, and Lisa Rybicki. "Symptom clustering in advanced cancer." Supportive care in cancer 14.8 (2006): 831-836.

[10] Huang M, Khurana A, Mastorakos GM, Fan J. Identification of Transportation Barriers in Patient Portal Messages via Deep Semantic Embeddings and Clustering. Stud Health Technol Inform. 2022 Jun 6;290:794-798. doi: 10.3233/SHTI220188. PMID: 35673127.

[11] Naleef Fareed, Daniel Walker, Cynthia J Sieck, Robert Taylor, Seth Scarborough, Timothy R Huerta, Ann Scheck McAlearney, Inpatient portal clusters: identifying user groups based on portal features, Journal of the American Medical Informatics Association, Volume 26, Issue 1, January 2019, Pages 28–36, https://doi.org/10.1093/jamia/ocy147

[12] “Nltk.chunk Package Documentation.” NLTK, https://www.nltk.org/api/nltk.chunk.html.

[13] Angelov, Dimo, "Top2Vec: Distributed Representations of Topics"

[14] Hossain, Dr. Shahriar. “Evaluation of Clustering Results.” Computing for All, Computing for All, 26 May 2021,https://computing4all.com/courses/introductory-data-science/lessons/evaluation-of-clustering-results/.

[15] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, Journal of computational and applied mathematics 20 (1987), 53-65

[16] “K-Means Clustering and Analysis.” Project Pro, https://www.projectpro.io/data-science-in-r-programming-tutorial/k-means-clustering-techniques-tutorial.

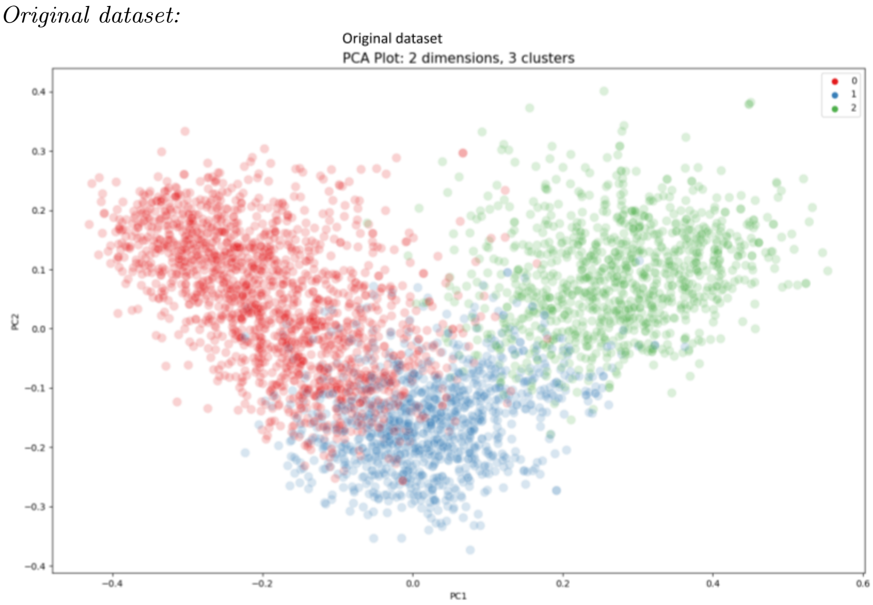
[17] https://github.com/MaartenGr/BERTopic

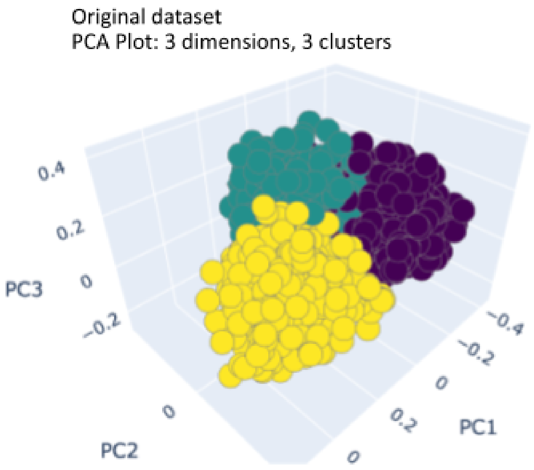
9 Appendix

9.1 Dimensionality-reduction technique plots

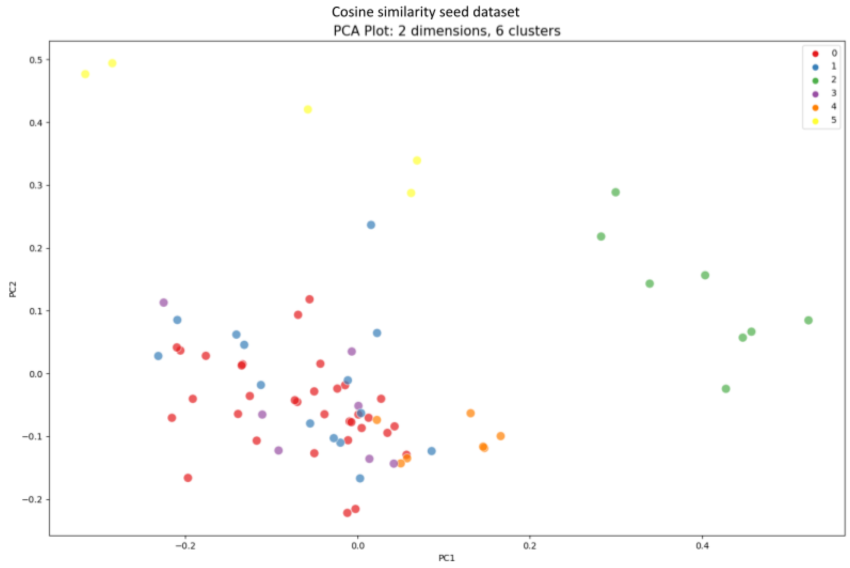
We provide both 2D and 3D versions of our PCA plots below.

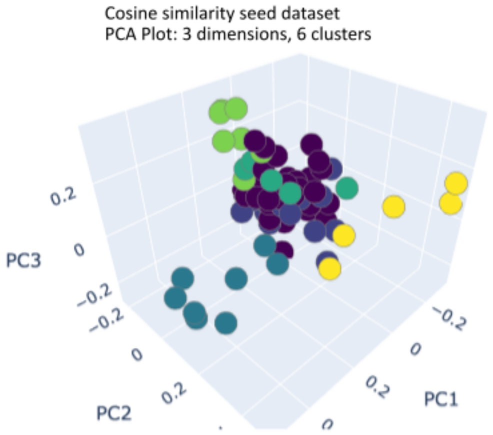
PCA Plots, 2D and 3D:
Original dataset:



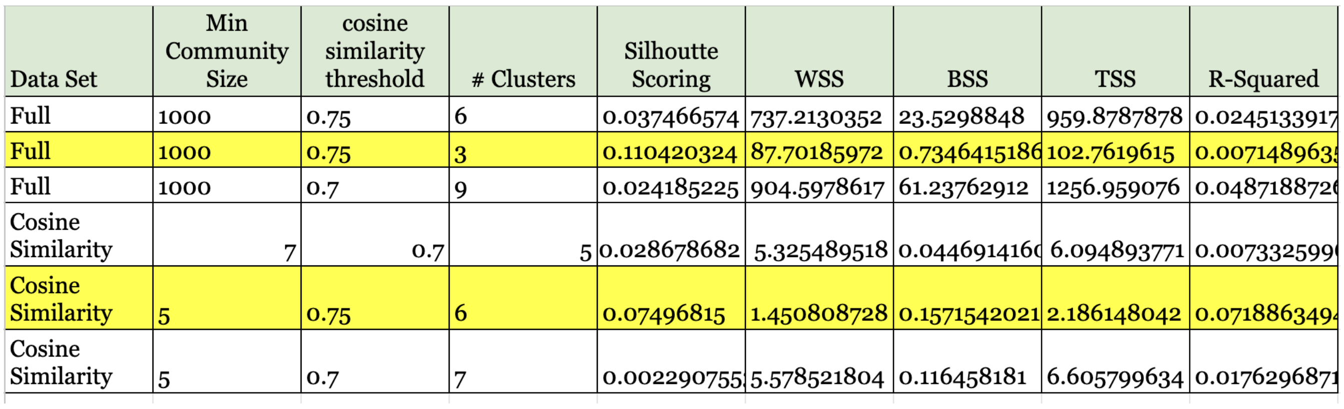


Cosine similarity seed dataset:





9.2 Additional Quantitative Data



Data Set	Min Community Size	cosine similarity threshold	# Clusters	Silhoutte Scoring	WSS	BSS	TSS	R-Squared
Full	1000	0.75	6	0.037466574	737.2130352	23.5298848	959.8787878	0.0245133917
Full	1000	0.75	3	0.110420324	87.70185972	0.7346415186	102.7619615	0.0071489635
Full	1000	0.7	9	0.024185225	904.5978617	61.23762912	1256.959076	0.0487188726
Cosine Similarity	7	0.7	5	0.028678682	5.325489518	0.0446914160	6.094893771	0.0073325990
Cosine Similarity	5	0.75	6	0.07496815	1.450808728	0.1571542021	2.186148042	0.0718863494
Cosine Similarity	5	0.7	7	0.002290755	5.578521804	0.116458181	6.605799634	0.0176296871

*Highlighted in Yellow Were Our Best Results
** Further experiments were taken, but this table is used to illustrate how with hyperparameter tuning we were able to optimize our clustering techniques