

# Discovering Related Clinical Concepts Using Large Amounts of Clinical Notes

## Supplementary Issue: Big Data Analytics for Health

Kavita Ganesan<sup>1</sup>, Shane Lloyd<sup>2</sup> and Vikren Sarkar<sup>2</sup>

<sup>1</sup>RxNLP, Salt Lake City, UT, USA. <sup>2</sup>Assistant Professor of Radiation Oncology, University of Utah School of Medicine and Huntsman Cancer Institute, Salt Lake City, UT, USA.

**ABSTRACT:** The ability to find highly related clinical concepts is essential for many applications such as for hypothesis generation, query expansion for medical literature search, search results filtering, ICD-10 code filtering and many other applications. While manually constructed medical terminologies such as SNOMED CT can surface certain related concepts, these terminologies are inadequate as they depend on expertise of several subject matter experts making the terminology curation process open to geographic and language bias. In addition, these terminologies also provide no quantifiable evidence on how related the concepts are. In this work, we explore an unsupervised graphical approach to mine related concepts by leveraging the volume within large amounts of clinical notes. Our evaluation shows that we are able to use a data driven approach to discovering highly related concepts for various search terms including medications, symptoms and diseases.

**KEYWORDS:** concept graph, related concepts, clinical concepts, data mining, knowledge discovery

**SUPPLEMENT:** Big Data Analytics for Health

**CITATION:** Ganesan et al. Discovering Related Clinical Concepts Using Large Amounts of Clinical Notes. *Biomedical Engineering and Computational Biology* 2016:7(S2) 27–33 doi:10.4137/BECB.S36155.

**TYPE:** Original Research

**RECEIVED:** February 9, 2016. **RESUBMITTED:** June 29, 2016. **ACCEPTED FOR PUBLICATION:** July 2, 2016.

**ACADEMIC EDITOR:** Kayvan Najarian, Editor in Chief

**PEER REVIEW:** Three peer reviewers contributed to the peer review report. Reviewers' reports totaled 1,132 words, excluding any confidential comments to the academic editor.

**FUNDING:** Authors disclose no external funding sources.

**COMPETING INTERESTS:** SL discloses membership on the marketing panel for Sirtex. Other authors disclose no potential conflicts of interest.

**COPYRIGHT:** © the authors, publisher and licensee Libertas Academica Limited. This is an open-access article distributed under the terms of the Creative Commons CC-BY-NC 3.0 License.

**CORRESPONDENCE:** ganesan.kavita@gmail.com

Paper subject to independent expert single-blind peer review. All editorial decisions made by independent academic editor. Upon submission manuscript was subject to anti-plagiarism scanning. Prior to publication all authors have given signed confirmation of agreement to article publication and compliance with all applicable ethical and legal requirements, including the accuracy of author and contributor information, disclosure of competing interests and funding sources, compliance with ethical requirements relating to human and animal study participants, and compliance with any copyright requirements of third parties. This journal is a member of the Committee on Publication Ethics (COPE).

Published by Libertas Academica. Learn more about this journal.

## Introduction

Related clinical concepts are two or more terms or phrases that are highly associated and related. For example, the term “nausea” is highly related to “vomiting” and the term “pregnancy” is highly related to “ectopic”. Such related concepts are extremely important to health-care-related applications. One important use case of *related concepts* is for hypotheses generation. Let us say, a physician is investigating a series of nausea events that have recently affected the local community. During this investigation, if he/she observes that the term “nausea” is being strongly associated with “salmon” and “salad” in the given time range of the event, the physician may choose to investigate this relationship further to understand if the salmon or the salmon in the salad was the root cause of those nausea events and take the necessary steps to resolve the issue. Another use case of *related concepts* is in medical and biomedical literature search. For example, if a user is looking for articles related to “opioid”, adding related concepts such as “analgesics” and “pain” to expand the query could further improve the relevance of the search results. Currently, in order to do this, most applications will have to rely on availability of manually curated concepts in published dictionaries. With these dictionaries however, there is no guarantee that the related concepts are in fact really related as it might include a related diagnosis or a parent concept instead of concepts that may occur together.

Most health-care applications today rely primarily on existing “static” terminologies curated by human experts such as SNOMED CT, LOINC, and RxNorm. While these controlled terminologies are extremely useful, they are highly dependent on human expertise, which leads to several issues. First, since these terminologies were hand curated by several subject matter experts from a specific geographic area, the actual terminologies used in practice can vary significantly from institution to institution let alone country to country. Thus, it does not account for locality of information. Another issue is that even though related concepts can be anything from symptoms to medications to procedures, each of these controlled terminologies will only show relationships explicitly defined by the expert based on his or her knowledge (eg, clinical findings with certain procedures only) and most likely would not be able to show concepts that can only be clinically observed. For example, a lookup on *chest pain* using the SNOMED CT browser returns concepts such as *dull chest pain*, *acute chest pain*, and *upper chest pain*, but does not show related concepts that one would find specifically in clinical notes such as *stuttering right-sided chest pain* or *radiating chest pain*. Note that both these statements appeared in the MIMIC II Clinical Database<sup>1</sup> that we used for the evaluation of our work. Furthermore, these clinical terminologies provide no form of evidence, score, or statistics that would be extremely



useful in applications. For example, what is the relationship strength between “chest pain” and “stuttering” or what is the probability of “chest pain” occurring with “vomiting” versus “chest pain” occurring with “acute”? Such questions cannot be answered with just controlled terminologies.

In this study, we thus explore a highly scalable graph-based approach to establish relationships between a *search query* and *related concepts* by leveraging large amounts of clinical notes. Specifically, we first construct a Concept-Graph using 10,000 clinical notes from the MIMIC II Database, where each node represents a unique word and the edges represent the links between the words. Then, given a query, we mine related concepts using links in the Concept-Graph and then rank the concepts found using a *relatedness* measure based on pointwise mutual information (PMI) and probability of co-occurrence. Evaluation on 10 different search topics, which include medications, symptoms, and diagnosis by five physicians, shows an average precision of 0.98 and a utility score above 0.90 using our best system.

One key advantage of our approach is that it is very general in that users would have complete control on the data used for constructing the Concept-Graph to find related concepts. For example, a user can construct a Concept-Graph using only cancer treatment related notes. A user may also choose to use all notes within the organization to construct a comprehensive Concept-Graph. Since the Concept-Graph also provides evidence information, one can directly obtain various statistical information from the graph to be used within an application. The resources used as part of this work can be found at <https://github.com/rxnlp/clinical-concepts>.

## Methods

The goal of this work is to find a list of *related clinical concepts* given a *search query* by leveraging large amounts of clinical notes. The intuition here is that the volume of clinical texts can provide hints on how related the concepts are to the *search query* based on a co-occurrence relationship within a specific distance. The *search query* can be any term such as a medication (eg, valium, lithium), disease (eg, COPD), side effect (eg, diarrhea, nausea), symptom (eg, chest pain, headache), or even a partial header name in a clinical note (eg, history). The *search query* can be a unigram (single word) or a multiword expression (eg, diabetes mellitus). The *related concepts* returned would be a list of unigrams ranked by their relatedness scores (we plan to explore multiword concepts in our future work). While there are many different ways to estimate the relatedness score between a *search query* and a candidate concept, in this paper, we evaluate two measures for relatedness, namely, a modified PMI measure and a probability of occurrence measure (PROB). We use the Concept-Graph data structure to efficiently represent large amounts of text in a way that enables quick lookup of statistical information based on the volume as well as provide indicators of which concepts are linked to the search query.

In the following subsection, we first describe how the Concept-Graph is constructed using large amounts of clinical

notes and the preprocessing involved in constructing the graph. Then, in “Discovering related concepts” section, we describe how the graph is used to find *candidate concepts* where some of these concepts become the *related concepts*. Finally, in “Computing relatedness scores for ranking related concepts” section, we describe the procedure that we use for relatedness scoring so that the related concepts can be ranked based on how relevant the concepts are to the search query.

**Concept-Graph construction.** The first step prior to building the Concept-Graph is to preprocess the clinical notes as they are fed into the Concept-Graph. We perform minimal preprocessing on the notes that include sentencings, lowercasing, and stop word removal. Each sentence in a clinical note is considered independent of one another. Sentences can be easily obtained from the clinical notes using existing sentence segmentation tools.<sup>2,3</sup> In our case, we developed a simple sentence segmenter using punctuation as heuristics. We also remove stop words from each sentence. Stop words are common words in the language that appear both in day to day language as well as very commonly in clinical notes. We appended the English stop words used within the Terrier Package<sup>4</sup> with some manually curated clinical stop words. The list of stop words used is published in <https://github.com/rxnlp/clinical-concepts>. While some of the common clinical note terms would naturally have a low rank using our system, these terms are distracting and yield unnecessary memory overhead and thus we dropped some of these words (eg, patient, clinic, and hospital). The preprocessing steps used in our work are graphically demonstrated in Figure 1.

Once preprocessing is complete, the Concept-Graph is constructed. The Concept-Graph is essentially an undirected positional word graph data structure that represents large amounts of natural language text in a compressed and easy to analyze format. It naturally models co-occurrence relationships between words, as each unique word is a node in the graph and the edges represent the relationship between words as it appears within sentences. This provides cues on which two concepts are related just by leveraging the links based on the original text. Each preprocessed sentence from the clinical notes is fed into the Concept-Graph data structure where each unique word becomes a node in the graph and each node holds the sentence identifier (SID) as well as position of the word in the sentence (PID). For example, if a word “asthma” appears 10,000 times in the text, there will only be a single node to represent asthma. The node representing asthma keeps track of which sentences used that particular word along with the corresponding positional information. An edge A-B is used to indicate that word “A” appeared at least once next to word “B” and the direction of the edge does not matter. Figure 2 shows an example of a Concept-Graph constructed with three preprocessed sentences from some clinical text. Notice that just based on this simple example, a strong link can already be seen for example between “vomiting” and “diarrhea”.

In constructing the Concept-Graph for this work, we utilized 10,000 clinical notes that were randomly picked from the

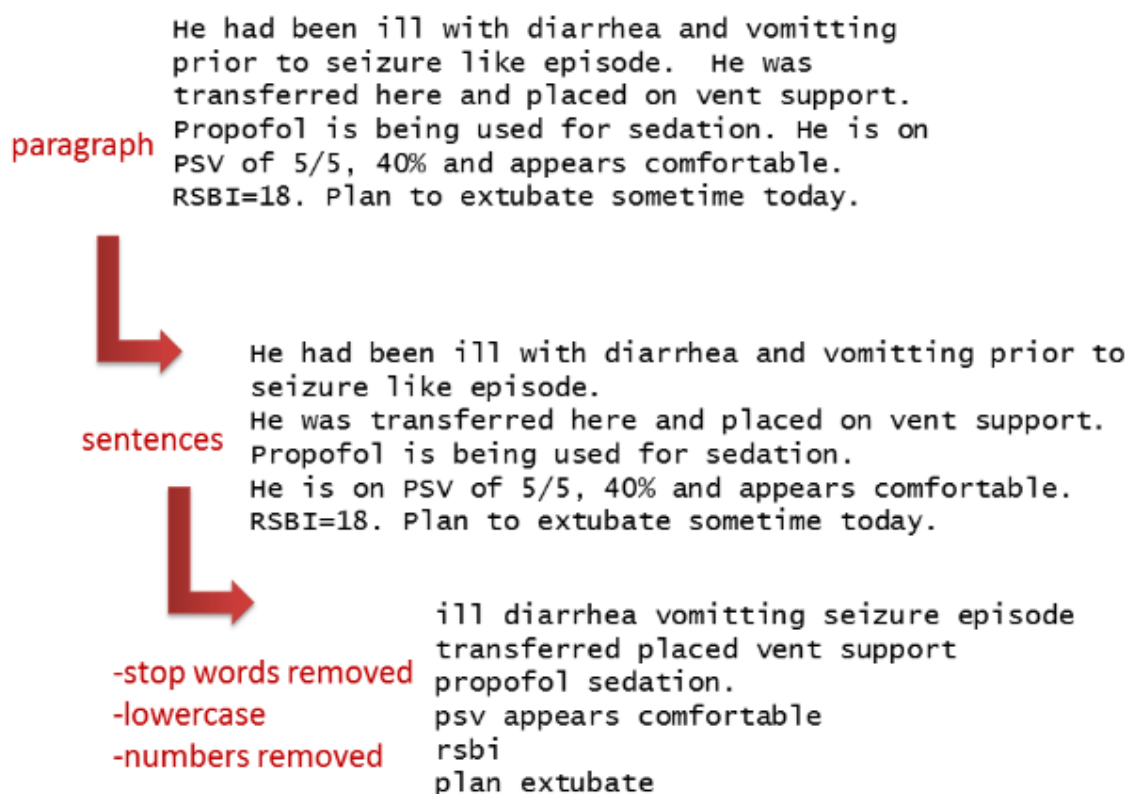


Figure 1. Preprocessing steps applied to clinical notes.

MIMIC II Clinical Database.<sup>1</sup> The MIMIC II Clinical Database contains comprehensive clinical data from intensive care unit (ICU) patients. These data were collected between 2001 and 2008 from different ICUs (medical, surgical, coronary care, and neonatal) in a single hospital. Thus, the query concept can be fairly general as the MIMIC II Database covers a range of treatments and conditions. We used all 10,000 notes to construct the Concept-Graph using the steps mentioned above.

**Discovering related concepts.** Once the Concept-Graph has been constructed, the next step is to find the related

concepts for a given query. For this, the query terms would first have to be identified in the Concept-Graph. If the query terms (eg, “chest” and “pain”) are themselves linked (there is an edge linking the terms), then all concepts that are linked to the query terms will first be identified. Based on Figure 2, for the query vomiting, linked concepts would include “morning”, “nausea”, and “diarrhea”. We will refer to these as *candidate concepts*. From these candidate concepts, we find the *related concepts* by eliminating weak links. Weak links are found by identifying candidate concepts that do not fulfill the minimum overlap requirements between the SIDs.

Specifically, if a query term,  $SQ$ , and a candidate concept,  $CC_i$ , share  $N$  sentences, the candidate concept is considered a *related concept* if the number of shared (overlapping) sentences exceed a threshold that we refer to as  $\sigma_{overlap}$ . In addition to this, the overlap computation is subject to a distance constraint referred to as  $\sigma_{window}$ , such that the distance between the positions of the words in consideration is no more than  $\sigma_{window}$ . For example, in Figure 3, based on just the overlap

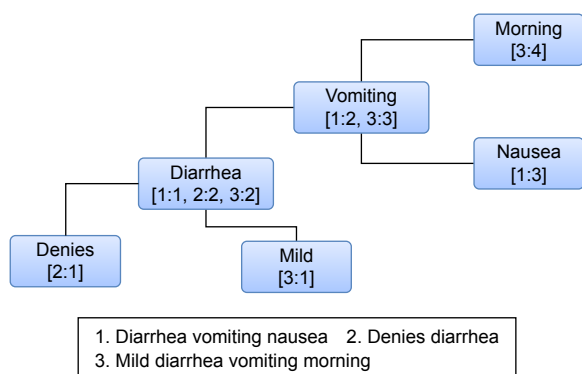


Figure 2. Example of Concept-Graph constructed using three sentences that have been preprocessed. Each node represents a unique word in the text. Each node stores the sentence identifier (SID) and the position of the word in the sentence (PID).



Figure 3. An overlap example, where vomiting is the query and nausea is the candidate concept. The format used is SID:PID where SID is the sentence identifier and PID is the position of the word in the sentence.



of the SIDs, vomiting and nausea have an overlap of 3 (SID: 3, 4, 5). However, when we take  $\sigma_{\text{window}} = 3$  into consideration, then the overlap will only be 2 (SID: 3, 4) since the distance between the position of the words for sentence 5, is 6 (5:12–5:6) and this exceeds the  $\sigma_{\text{window}}$  threshold. Thus, if we use  $\sigma_{\text{overlap}} = 3$  and  $\sigma_{\text{window}} = 3$ , then this relationship will be considered a weak link since the overlap ends up being only 2.

The intuition for the  $\sigma_{\text{window}}$  restriction is that words that appear much further away are less likely to be related to the query term than words that are much closer in general to the search term. Formally, the overlap between a search query, SQ, and a candidate concept,  $CC_i$ , can be expressed as follows:

$$\text{Overlap}(\text{SQ}, \text{CC}_i) = \text{count}(\text{SID}_{\text{SQ}} \cap' \text{SID}_{\text{CC}_i}) \quad (1)$$

where  $\text{SID}_{\text{SQ}}$  contains all the SIDs that pass through the search query and  $\text{SID}_{\text{CC}_i}$  are all the sentences that pass through a candidate concept. The  $\cap'$  refers to overlap between the SIDs where the positional difference between the overlapping sentences are no more than  $\sigma_{\text{window}}$ . If  $\text{Overlap}(\text{SQ}, \text{CC}_i) < \sigma_{\text{overlap}}$ , then this candidate concept will be discarded from consideration as it is regarded as a weak link.

**Computing relatedness scores for ranking related concepts.** While a query term may be linked with thousands of related concepts, there are some concepts that are more related than others. For example, for the query vomiting, intuitively we know that the term nausea is more strongly related to vomiting than the word morning. The term morning most likely appears in the context of morning sickness where a patient experiences vomiting. Another example is for the query asthma where the medication “QVAR” is more related to asthma than the word medication itself. Thus, to distinguish concepts that are highly related from ones that are marginally related, we introduce a ranking system that ranks relationships based on a *relatedness* score. Given a search query, SQ, and a related concept,  $RC_i$ , we denote the relatedness score as  $\text{Relatedness}(\text{SQ}, \text{RC}_i)$ . We evaluate two different ways to compute  $\text{Relatedness}(\text{SQ}, \text{RC}_i)$  with the first measure estimating the likelihood of a related concept occurring with the search query as follows:

$$\text{Relatedness}(\text{SQ}, \text{RC}_i)_{\text{PROB}} = \log_2 \left\{ \frac{\text{Overlap}(\text{SQ}, \text{RC}_i)}{\text{SID}_{\text{SQ}}} \right\} \quad (2)$$

where  $\text{SID}_{\text{SQ}}$  is the total number of sentences containing the search query and  $\text{Overlap}(\text{SQ}, \text{RC}_i)$  can be computed using Equation 1.  $\text{Relatedness}(\text{SQ}, \text{RC}_i)_{\text{PROB}}$  is simply the probability that the search query and related concept occur together within a window of  $\sigma_{\text{window}}$ . The second measure we used is a modified PMI measure,<sup>5</sup> which measures the strength of two concepts occurring together versus the two

concepts occurring independently. The  $\text{Relatedness}(\text{SQ}, \text{RC}_i)$  score based on PMI is measured as follows:

$$\text{Relatedness}(\text{SQ}, \text{RC}_i)_{\text{PMI}} = \log_2 \left\{ \frac{P(\text{RC}_i \cap \text{SQ})}{P(\text{RC}_i) * P(\text{SQ})} * \log_2(X) \right\} \quad (3)$$

where  $X$  is computed based on Equation 1 as:

$$X = \text{Overlap}(\text{SQ}, \text{RC}_i) \quad (4)$$

The first part of Equation 3,  $\frac{P(\text{RC}_i \cap \text{SQ})}{P(\text{RC}_i) * P(\text{SQ})}$  computes the actual strength between the search query and the related concept. The multiplication with  $\log_2(X)$  is to account for frequency of occurrence. Without the  $\log_2(X)$ , the equation will assign a very high score to concepts that rarely appear together. For example, these can be noisy concepts such as “vomiting” and “naesa”, which is a misspelling that may occur just once or twice together in the entire corpus. By taking the frequency count into consideration, some of the false positives can be eliminated.

Once  $\text{Relatedness}(\text{SQ}, \text{RC}_i)_{\text{PROB}}$  and  $\text{Relatedness}(\text{SQ}, \text{RC}_i)_{\text{PMI}}$  are computed for each related concept for a given search query, these related concepts are sorted by decreasing order of their relatedness scores.

## Evaluation

The goal of our evaluation is to understand if the top related concepts discovered by our system are in fact related and relevant to the search query. For example, is “vomiting” related to “nausea”? To perform our evaluation, we first requested a physician to provide 10 search queries that he/she may want to search for to find related concepts. The topics we received were: lithium (medication), beta-blocker (medication), penicillin (antibiotic medication), advair (inhaler for asthma treatment), chest pain, pregnancy, myocardial infarction, bloody stool, fracture, and syncope.

**Ground truth.** We then used our system to generate the top 50 related concepts for these 10 topics using the two relatedness measures described in “Methods” section. We then presented the results to five independent physicians to rate our results. We asked these five physicians to rate the results based on the two rankings (PMI and PROB) as follows:

**R: Relevant.** If the query concept and the related concepts are highly related, the related concepts returned do not have to be diagnosis related but it can be a symptom, a medication, a condition, or a side effect. Anything specific that is highly related to the main search term (eg, chest pain and “palpitations”).

**RG: Relevant but general.** If the related concept is relevant to the query concept but the related concept is quite general (eg, chest pain and “denies”).

**N: Noise.** If the related concept is noise such as a common English word or the query and related concept should never occur together.

**Table 1.** Precision of top 50 results.

SEARCH TERM	PMI	PROB
Chest pain	0.980	0.918
Syncope	0.980	0.959
Lithium	0.939	0.959
Advair	1.000	1.000
Myocardial infarction	1.000	0.980
Bloody stool	0.980	0.980
Beta-blocker	0.980	0.939
Fracture	0.980	1.000
Penicillin	0.959	0.959
Pregnancy	1.000	0.898
Average	<b>0.980</b>	<b>0.959</b>

Given the ratings from the five physicians, we used majority vote as the final rating. This means that if a related concept has been rated as **R** by three physicians and **RG** by two physicians, we use **R** as the rating. In a situation where there is a tie, we use the more conservative of the ratings that are a tie. So, if two physicians rated **RG**, two rated **N**, and one rated **R**, we use **N**. Similarly, if two raters rated **RG**, two rated **R**, and one rated **N**, we end up using **RG**.

**Evaluation metric.** To evaluate the overall performance of our system, we introduce two measures, one being precision and the other we refer to as a utility score. Precision evaluates how many of the top 50 concepts are relevant (ie, not noise). Utility score on the other hand assigns a score for each type of concept produced. Specifically, a score of 2 is assigned if the system found a relevant concept (**R**), a score of 1.5 if the system produced a relevant but general concept (**RG**), and a score of  $-2$  if the system produces noise. With this, the more noise the system produces, the more it gets penalized and the more relevant concepts the system produces the better the overall score. Given top  $N$  concepts, the utility score is computed as follows:

$$\text{Utility score (results)} = \frac{\sum_{i=1}^N \text{utility (concept}_i)}{2.0}$$

where  $\text{utility (concept}_i)$  refers to the utility score assigned to each related concept and  $N$  is the maximum number of

concepts in consideration. The denominator value of 2.0 is the maximum possible utility score. With this, Utility score (results) will have a score within the range of  $[-1, 1]$  where scores that are closer to  $-1$  indicate that the results are very noisy and the opposite can be inferred when the scores are closer to 1. We do not measure recall in this paper since the related terms discovered are data dependent and the better the quality of input data set and the larger the volume, the higher the possibility of recovering meaningful concepts.

## Results

**Precision of results.** We first look into precision of our results to understand how many nonrelevant concepts are produced by the system. Table 1 shows a summary of precision based on our ground truth. Notice that with both rankings, the amount of noise produced is extremely low with above 95% of the results being either **R** or **RG**. In fact, the PMI-based ranking has an average precision of 98%. This indicates that all in all, the system finds concepts that are related to the search query. Also notice that in general, the overall precision of the PMI-based ranking is slightly higher than that of the PROB-based ranking. This shows that overall, PROB-based ranking introduces more noise than PMI.

**Utility of results.** The utility of results indicate how usable the top related concepts are in practice. The less relevant the top concepts, the lower the utility score at different rank cutoffs. Based on Table 2, we can see that overall the PMI-based ranking provides a higher utility compared with the PROB-based ranking. This is because the PMI-based ranking immediately returns concepts that are considered **R** at the very top as opposed to PROB-based ranking where the top concepts are sometimes rated **RG** and **N** in some cases. As more and more related concepts are taken into consideration (from lower ranks), the difference in utility scores becomes slightly smaller between the two rankings (see  $\text{Utility}@50$ ). Since PMI-based ranking is able to immediately return concepts that are most relevant to the search query, it would be more effective in practice for tasks such as query expansion and hypothesis generation.

**Sample results.** Table 3 shows a snapshot of the top related concepts for four different search queries based on PROB-based ranking and PMI-based ranking. Notice that with the PMI-based rankings, the top related concepts are very specific to the search query. With the PROB-based ranking, the top concepts are more general and the top PMI concepts

**Table 2.** Utility scores at different rank cutoffs.

	UTILITY@5	UTILITY@10	UTILITY@20	UTILITY@30	UTILITY@50
PROB	0.848	0.872	0.862	0.851	0.816
PMI	0.945	0.943	0.935	0.913	0.900
Difference	+11.50%	+8.18%	+8.48%	+7.29%	+9.10%

**Notes:** @ indicates the number of related concepts in consideration. The maximum possible utility score is 1 and the lowest possible utility score would be  $-1$ .

**Table 3.** Top 15 related concepts for four search queries ranked based on PMI and PROB scores.

PMI-BASED RANKING	SCORE	PROB-BASED RANKING	SCORE
<b>Chest pain</b>			
Pleuritic	14.1634	Breath	-3.1421
Substernal	13.4485	Shortness	-3.1595
Ictus	12.9084	Denies	-3.8398
Palpitations	12.8769	Year	-4.4803
Radiating	12.8751	Cath	-4.5945
Pleuric	12.7979	Sob	-4.6244
Anomolous	12.7943	Pleuritic	-4.6458
Aur	12.6486	Reason	-4.7249
Epigatric	12.6055	Man	-4.8863
Squeezing	12.5825	Back	-5.0036
Tightness	12.5236	Major	-5.0598
Shortness	12.5001	Surgical	-5.0639
Experience	12.3301	Left	-5.1356
Stuttering	12.3132	Sided	-5.1442
Crushing	12.0798	Abdominal	-5.2019
<b>Syncope</b>			
Presyncope	16.5897	Year	-3.0528
Vasovagal	14.4518	Telemetry	-3.1863
Orthopnea	13.9959	Reason	-3.4546
Palpitations	13.9798	Man	-3.7325
Telemetry	12.6358	Woman	-3.7713
Nocturnal	12.5801	Pain	-4.0291
Lightheadedness	11.848	Contrast	-4.0291
Dyspnea	11.2657	Presyncope	-4.0932
Near	10.6774	Palpitations	-4.1776
Ankle	10.5456	Fall	-4.1776
Suffered	10.4894	Dyspnea	-4.4866
Tias	10.3803	Near	-4.5303
Episode	10.2679	Episode	-4.5303
Paroxysmal	10.135	Chest	-4.5754
Fall	9.9611	Orthopnea	-4.7198
<b>Advair</b>			
Diskus	18.804	Bid	-2.0538
Spiriva	17.3586	Albuterol	-2.3352
Discus	17.3257	Diskus	-2.9652
Singulair	15.5246	Puff	-3.0721
Puff	15.2113	Daily	-3.1096
Inh	14.3914	Prn	-3.2701
Tiotropium	13.7217	Mcg	-3.4996
Inhaler	13.6754	Spiriva	-3.6027
Albuterol	13.0639	Dose	-3.6571
Disk	12.9482	Nebs	-3.7137
Puffs	12.5906	Combivent	-3.834

**Table 3.** (Continued)

PMI-BASED RANKING	SCORE	PROB-BASED RANKING	SCORE
Combivent	12.4954	Medications	-3.834
Zocor	12.3302	Discharge	-3.834
Flovent	12.2347	Inhaler	-4.1876
Nebs	11.9469	Puffs	-4.1876
Qday	11.7117	Disk	-4.2701
<b>Fracture</b>			
Comminuted	13.0576	Left	-3.0121
Dislocation	12.7384	Right	-3.0912
Nondisplaced	12.6857	Evidence	-3.4714
Malalignment	12.6517	Comminuted	-4.0818
Displaced	12.4208	Dislocation	-4.1051
Intertrochanteric	12.3706	Acute	-4.399
Diaphyseal	12.3344	Distal	-4.505
Subcapital	12.2103	Displaced	-4.5227
Burst	12.1787	Compression	-4.7435
Fibular	12.1329	Seen	-4.796
Styloid	12.0737	Rib	-4.797
Midshaft	12.0667	Spine	-4.8751
Radius	12.0599	Identified	-4.9203
Acetabular	12.0419	Report	-4.9692
Subtrochanteric	12.0087	Final	-4.9785

start appearing in the later ranks of PROB-based ranking. This is interesting because it produces two use cases for applications. For example, some applications may value related concepts that are more general in which case the PROB-based ranking would be more suitable. Some other applications may value concepts that are very clinically related, and in that case, the PMI-based ranking would be ideal.

### Example Usage in Practice

The concept-graph can be used in a variety of settings once it has been constructed. For example, in medical literature search, the query terms used by the user can be expanded with related concepts to improve the search results. In more specific terms, if the user is interested in literature related to asthma, related concepts such as “advair”, “shortness”, and “breath” could help bring up literature that is more relevant. Another example is in the case of ICD-10 code set filtering. If a particular ICD-10 code (ie, the description of the code after stop word removal) matches none of the top  $N$  concepts related to the clinical finding or procedure in question, then the ICD-10 code suggested can be disregarded. For instance, let us say, the clinical finding is chest pain and the suggested ICD-10 code is “M79.642 Pain in left hand”. After stop word removal, this becomes “pain”, “left”, and “hand”. While pain and left would match related concepts of chest pain, none of the top related concepts for chest pain would



be a match to hand. Thus, this code can be eliminated from the list of suggestions. This can improve precision of ICD-10 code set suggestions within automatic ICD-10 coding systems.

## Conclusion

In this work, we proposed a method to mine related clinical concepts by leveraging the volume within large amounts of clinical notes along with a graph data structure. Our evaluation shows that our system is able to return highly relevant concepts with above 95% precision and our best method achieves an average utility score of 0.90. This shows that the related concepts generated by our system can be immediately used for a variety of tasks, including query expansion, hypothesis generation, incident investigation, sentence completion, and ICD-10 code set filtering.

Our system is not only lightweight wherein it relies on limited linguistics resources but also very general in that the same method can be applied to different types of big clinical data. The only requirement for our method to work is to have *volume* in data, which is almost not a problem in this era of Big Data. For example, we can run our method on all clinical notes from a specific department (eg, cardiology) across different organizations to obtain a very focused set of related concepts. We can even use the same method on all clinical notes within a particular time range to investigate an incident or an outbreak.

While this work was evaluated using the MIMIC II Database that is a fairly general dataset, we would like to explore its use in a more narrow situation to understand its

applicability in a clinical setting. We would like to work with a physician in actually using our system to investigate certain surprising relationships that could help in their future clinical investigation.

## Author Contributions

Conceived and designed the experiments: KG, SL and VS. Analyzed the data: KG, SL and VS. Wrote the first draft of the manuscript: KG, SL and VS. Contributed to the writing of the manuscript: KG, SL and VS. Agree with manuscript results and conclusions: KG, SL and VS. Jointly developed the structure and arguments for the paper: KG, SL and VS. Made critical revisions and approved final version: KG, SL and VS. All authors reviewed and approved of the final manuscript.

## REFERENCES

1. Saeed M, Villarroel M, Reisner AT, et al. Multiparameter intelligent monitoring in intensive care ii (mimic-ii): a public-access intensive care unit database. *Crit Care Med.* 2011;39(5):952.
2. Matusov E, Mauser A, Ney H. Automatic sentence segmentation and punctuation prediction for spoken language translation. In: IWSLT. Kyoto, Japan; 2006:158–165.
3. Palmer DD. Tokenisation and sentence segmentation. *Handbook of Natural Language Processing*. Edited by Robert Dale, Harold Somers, and Hermann Moisl. Boca Raton, FL: CRC Press; 2000:11–35.
4. Ounis I, Amati G, Plachouras V, He B, Macdonald C, Lioma C. *Terrier: A High Performance and Scalable Information Retrieval Platform*. Proceedings of the OSIR Workshop, Seattle, 2006.
5. Ganesan K, Zhai C, Viegas E. Micropinion generation: An unsupervised approach to generating ultra-concise summaries of opinions. In: Proceedings of the 21st International Conference on World Wide Web, WWW'12. New York, NY: ACM; 2012:869–878.