A Matching Framework for Modeling Symptom and Medication Relationships from Clinical Notes

Yuan Ling, Yuan An
College of Computing & Informatics
Drexel University
Philadelphia, USA
{yl638, ya45}@drexel.edu

Xiaohua Hu
College of Computing & Informatics
Drexel University
Philadelphia, USA
xh29@drexel.edu

***Abstract*—Clinical notes are rich free-text data sources containing valuable symptom and medication information. Little research has been done on matching medication information with multiple symptoms information. Such a matching could provide valuable information for patients with multiple syndromes. We propose a Symptom-Medication (Symp-Med) matching framework to model symptom and medication relationships from clinical notes. After extracting symptom and medication concepts, we construct a weighted bipartite graph to represent the relationships between the two groups of concepts. The key is to efficiently answer user's symptom-medication queries using the graph. We formulate this problem as an Integer Linear Programming (ILP) problem. The objectives are to maximize the total edge weight and minimize the number of medication concepts. We first explore a Branch-and-Cut based algorithm. Then, we revise the combinational objective, and propose a Greedy-based algorithm for solving the Symp-Med problem. The Greedy-based algorithm performs better and significantly improves the computational costs.**

Keywords—Symp-Med Matching Framework, Symptom, Medication, Clinical Notes

I. INTRODUCTION

Clinical Narratives contain a lot of valuable information about patients, such as medication conditions (diseases, injuries, medical symptoms, and etc.) and responses (diagnoses, procedures, and drugs) [1]. These types of valuable information extracted from clinical narratives can be used to build profiles for individual patients [2], discover disease correlations [3] and enhance patient care [4].

Symptoms and medications are two important types of information that can be obtained from clinical notes. Symptom information such as diseases, syndromes, signs, diagnose etc., can be used to analyze diseases for patients. We define this symptom information as symptom concepts in this paper. In addition, valuable medication information is commonly embedded in unstructured text narratives spanning multiple sections in medical documents [5]. Medication information from clinical notes is often expressed with medication names and other signature information about drug administration, such as dosage, route, frequency, and duration. In this paper, we extract medication names from clinical notes, and use medication names as medication concepts. Other related medication information is also very important, and will be considered in future research.

Currently, large volumes of clinical documents are generated by electronic health record systems [6]. On one hand, these clinical documents are unstructured or semi-structured. It is a difficult task to extract information from these documents. Symptom information and medication information extraction for clinical notes need sophisticated clinical language processing methods [7]. On the other hand, due to the individual diversity, discovering and mining relationship between symptom information and medication information from clinical texts becomes a challenge problem. These underutilized resources have a huge potential to improve health care. It is very important for patients with multiple syndromes to learn the relationships between symptoms and medications as indicated in the scenario below.

A use case scenario: a new patient is diagnosed with alcoholic liver disease (ALD) and type2 diabetes. A set of related symptoms are observed, so a set of medications should be prescribed to treat these symptoms. In the meantime, related clinical notes extracted from a database with symptoms and medications highlighted will also be presented as evidences to the physician and patient. The physician can use these clinical notes to support decisions, and the patient might find the medications given by physician more convincing based on the clinical notes from other patients who had similar medical conditions.

In this paper, we study the following questions: How to represent the relationship of symptom concepts and medication concepts we extracted from clinical notes? How to extract a set of most valuable medication concepts for a patient with a set of known symptom concepts? To the best of our knowledge, little previous work has systematically studied these problems.

The rest of the paper is organized as follows: Section II introduces clinical notes as background, and presents the overview of how we extract symptom concepts and medication concepts from clinical notes. Section III proposes a Symp-Med framework, and defines a weight matrix for the framework. Section IV formalizes our problem, and implements the Symp-Med matching algorithms. Section V presents our experiment dataset, evaluation methodology, and results. Section VI

discusses related research work. Section VII presents our conclusions and future work.

II. Information Extraction From Clinical Notes

Clinical Note is an important part of patient records in an unstructured free-text format. Symptom and medication concepts are valuable information, which are embedded in multiple sections in clinical note.

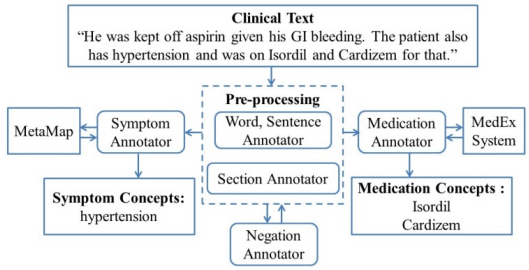


Fig. 1. An overview of symptom/medication extraction from Clinical Notes

We extract symptom concepts and medication concepts from clinical notes for our framework. An overview of extracting symptoms and medications from clinical notes is showed in Fig. 1. We extract the symptom concepts such as "hypertension" and medication concepts such as "Isordil, Cardizem" from the clinical texts "He was kept off aspirin given his GI bleeding. The patient also has hypertension and was on Isordil and Cardizem for that."

First, we pre-process clinical notes to identify words and sentences from clinical notes using Stanford CoreNLP Tool (http://nlp.stanford.edu/downloads/). During the pre-processing, we use section annotator to identify different sections for each clinical note. The section annotator depends on the section header information from clinical notes. Negation sections, such as "ALLERGIES" or "Family History", are excluded. For example, "She is allergic to MORPHINE" from the section "ALLERGIES", medication name "MORPHINE" is a negation medication name, so we exclude it.

We also use negation annotator to remove negation symptom and medication concepts. An example is that "The patient was told to avoid taking aspirin or any other NSAIDs given his GI bleed", we remove "aspirin" and "NSAIDs" because of the pre-negation words "avoid". Pre-negation and post-negation are defined in Negation maker (NegEx: http://www.dbmi.pitt.edu/chapman/NegEx.html). Pre-negation is negation words like avoid, deny, cannot, without, and so on. Post-negation is negation words like free, was ruled out, and so on.

After pre-process, we use symptom annotator based on the MetaMap [8] to extract symptom concepts from clinical notes. Meanwhile, we use medication annotator based on MedEx System [9] to extract medication concepts from clinical notes.

We use MetaMap to extract symptom concepts from clinical notes. MetaMap (http://nls3.nlm.nih.gov) is a program that maps biomedical texts to concepts in the UMLS Meta-thesaurus [8, 10]. Since Metamap returns all types of concepts, we only keep these concepts related to symptoms, such as concept labeled as "sosy", which represents "sign and symptom". The related types of concepts include: {sosy, dsyn, neop, fngs, bact, virs, cgab, acab, lbtr, inpo, mobd, comd, anab}, see [11] in detail.

We use MedEx system to extract medication concepts from clinical notes. The MedEx system is a natural language processing system to extract medication information from clinical notes [9]. In clinical notes, medication data are often expressed in medication names and signature information about drug administration. The MedEx system extracts multiple semantic categories of medication findings from clinical notes, such as DrugName, Strength, Route, Frequency, Form, Dose Amount, IntakeTime, Duration, Dispense Amount, Refill, and Necessity. Here we use the DrugName as medication concept.

III. Symp-Med Framework

Base on the symptom concepts and medication concepts extracted from clinical notes, we develop a Symp-Med Framework. The major component of this framework is a Symptom and Medication Bipartite Graph (Symp-Med Bi-graph).

A. Symp-Med Graph

The Symp-Med Bi-graph is a bipartite graph $G = \langle S \cup D, E \rangle$. There are two groups of nodes S and D. There is no edge between vertices in the same group. S is a set of vertices representing symptom concepts from clinical notes, $S = \{s_i | 1 \le i \le p\}$. D is a set of vertices representing medication concepts from clinical notes, $D = \{d_i | 1 \le i \le q\}$. E is a set of edges between the vertices from D and S, $E \in S \times D$. M is a set of weights representing weight value for each edge in set E.

The Symp-Med Bi-graph G can be represented by a $p \times q$ dimension matrix M, where m_{ij} is the weight value of edge $< s_i, d_j >$. For each clinical note, we use the symptom and medication concepts to form a matrix M. We set the value of m_{ij} based on the relation information we extracted from the clinical note. We aggregate all matrix M for individual clinical notes (in the clinical note level) to form a new matrix W for all clinical notes (in the cluster level).

B. Weight Matrix Definition

For a clinical note, we extract a set of symptom concepts $S = \{s_i | 1 \le i \le p\}$ and a set of medication concepts $D = \{d_j | 1 \le j \le q\}$. A matrix $M_{p \times q}$ can be built based on these two sets of concepts. We define a weight factors set $F = \{f^r | 1 \le r \le k\}$, which contains multiple weight factors. The weight factor set decides the weight values for each concepts pair $\langle s_i, d_j \rangle$. Weight values represent the relevance between symptom concept and medication concept. The larger the weight values, the more relevant the two concepts are. The weight value m_{ij} for concept pair $\langle s_i, d_j \rangle$ with weight factor value is defined as follows

$$m_{ij} = \sum_{r=1}^{k} f_{ij}^{r} \qquad (1)$$

We define two weight factors for Eq. 1 in this paper. One is a "Co-occurrence" factor f_{ij}^{1}. If symptom concept s_i and medication concept d_j appear in the same clinical note, $f_{ij}^{1} = 1$.

Otherwise, $f_{ij}^1 = 0$. The second weight factor is a "Co-occurrence in the same section" factor f_{ij}^2. If symptom concept s_i and medication concept d_j appear in the same section of a clinical note, $f_{ij}^2 = 1$. Otherwise, $f_{ij}^2 = 0$.

For all clinical notes $C = \{c_i | 1 \leq i \leq k\}$, a matrix W for all clinical notes C is constructed by integrating all weight matrixes M.

IV. Symp-Med Matching Algorithm

In the weight matrix W learned from the Symp-Med framework, the weight values represent the relevance relations between symptom concepts and medication concepts. For the Symp-Med framework, we define the Symp-Med matching problem. For a set of symptom concepts from a patient as the input, we want to predict a set of medication concepts as the output with the maximized total edge weight value and minimized number of medications. A motivating example for our Symp-Med matching problem is illustrated as follows.

A patient has two symptoms: fever and runny nose. A physician may have two kinds of prescriptions for this patient. The first prescription contains one medication, "Compound Paracetamol and Amantadine Hydrochloride Tablets". The second prescription contains two medications, "Acetaminophen" and "Nasal Drops". Suppose the first prescription has a higher weight value with these two symptoms than the second prescription. First, set 1 (Compound Paracetamol and Amantadine Hydrochloride Tablets), and set 2 (Acetaminophen and Nasal Drops) should be matched as two medication sets for these two symptoms. Second, since the first prescription "Compound Paracetamol and Amantadine Hydrochloride Tablets" has the larger weight value and smaller number of medications, it should be matched as the top one in the output set.

A. Symp-Med Matching Problem Formulation

We formulate the Symp-Med matching problem as follows.

1) Input

For this Symp-Med matching problem, the input includes a weight matrix W and a query vector S'. The weight matrix W is a $m \times n$ dimension matrix. The matrix describes the weight values of relevance edges between a set of symptom concepts $S = \{s_1, \dots, s_m\}$ and a set of medication concepts $D = \{d_1, \dots, d_n\}$. The query vector S' is described as follows

$$S' = \{s'_1, \dots, s'_p\},$$

$$p \leq m,\ S' \subseteq S,\ \text{where } i, j \in \{1,2,\cdots p\}, \text{ and } \forall i \neq j, s'_i \neq s'_j$$

2) Output

Given the weight matrix W and query vector S', we want to get a set of medication concepts as output, which can be represented as a vector as follows

$$D' = \{d'_1, \dots, d'_q\},$$

$$q \leq n,\ D' \subseteq D, \text{ where } i, j \in \{1,2,\cdots, q\}, \text{ and } \forall i \neq j, d'_i \neq d'_j$$

3) Constraints and Goal

The solution is a sub matrix of W for the query vector S' and the output vector D'. This sub matrix W' is $p \times q$ dimension matrix. In order to guarantee that the summation value of all elements from one row in matrix W' is bigger than zero, a constraint is set as follows

$$\sum_{j \in \{1,\dots,q\}} w'_{ij} > 0, \text{ for any } i \in \{1, \cdots, p\} \qquad (2)$$

That means there is at least one element larger than zero in each row since all weight values are either equal to zero or larger than zero.

The goal of this problem is two-fold:

First, maximize the sum of all elements (total weight value) from Matrix W', which is described as follows

$$\sum_{i \in \{1,\dots,p\}, j \in \{1,\dots,q\}} w'_{ij}$$

Second, minimize the number of columns q. That means the size of output vector should be as small as possible.

B. Symp-Med Matching Algorithm

First, the Symp-Med matching problem can be formulated as an ILP problem, the form of this ILP problem is described as follows

$$\textit{Maximize } \sum_{i=1}^{p} \sum_{j=1}^{n} w_{ij} z_{ij} - \varepsilon \sum_{j=1}^{n} y_j$$

$$\textit{Subject to}$$

$$\sum_{j=1}^{n} z_{ij} \geq 1,\ \forall i \in \{1, \dots, p\}$$

$$z_{ij} \leq y_j,\ \forall i \in \{1, \dots, p\},\ \forall j \in \{1, \dots, n\}$$

$$z_{ij} \in \{0,1\}^{p \times n}$$

$$y_j \in \{0,1\}^n \qquad (3)$$

Eq.3 uses z_{ij} and y_j to decide whether an element $d_j \in D$ should be selected to D' or not. $z_{ij} = 1$ means that the edge $< s_i, d_j >$ is selected. $z_{ij} \leq y_j$, $\forall i \in \{1, \dots, p\}$, $\forall j \in \{1, \dots, n\}$ means if any edge connect with d_j is selected, d_j need to be selected. $y_j = 1$ represents d_j is selected. If none of edges connecting to d_j is selected, d_j is not selected, then $y_j = 0$.

ε is a parameter to balance the two objectives $\sum_{i=1}^{p} \sum_{j=1}^{n} w_{ij} z_{ij}$ and $\sum_{j=1}^{n} y_j$. ε is set dynamically as follows

$$\varepsilon = \varepsilon' \times \max_i\left(\sum_{j=1}^{n} w_{ij}\right), i \in \{1, \dots, p\},\ \varepsilon' \in (0, 1],$$

$\varepsilon' = 1$ represents minimizing $\sum_{j=1}^{n} y_j$ as much as possible, if constraints are all satisfied, no extra d_j will be selected. If $\varepsilon' > 1$, the result is the same as the result when $\varepsilon' = 1$. The decrease of ε' from one to zero will improve the number of selected d_j. When $\varepsilon' = 0$, the minimizing objective $\sum_{j=1}^{n} y_j$ is not considered. In order to take the maximized total weight value and the minimized selected d_j number both into consideration, ε' is set as $\varepsilon' \in (0, 1]$.

The ILP problem formulated in Eq.3 is an NP-hard problem. Approximation algorithms are developed for dealing with ILP problem, such as Primal-Dual method [12], and Linear Programming (LP) relaxation and rounding method.

Here we use a branch-and-cut algorithm [13] to solve the ILP problem. The branch-and-cut algorithm is implemented in GLPK MIP solver [14].

Alg. 1: Branch-and-Cut based Symp-Med Matching
input: weight matrix $W \in R^{p \times n}$, parameter ε'
output: vector D'
begin
  Let ILP_{SMM} be the linear integer programming formulation as Eq.3
  $Y \leftarrow branch\_and\_cut(ILP_{SMM})$
  for $y_j \in Y$ **do**
    if $y_j = 1$ **then**
      $D' = D' \cup \{d_j\}$
    end if
  end for
Return: D'
end.

The branch-and-cut algorithm needs to relax the $\mathrm{ILP_{SMM}}$ to a corresponding $\mathrm{LP_{SMM}}$. The computational effort to solve LP is bounded by a polynomial function of problem size. The problem size of this $\mathrm{LP_{SMM}}$ is $(p+1)n$. A possible computational complexity is $O(pn^2)$ [15].

Since the two objectives in the Symp-Med matching problem are maximizing $\sum_{i=1}^{p}\sum_{j=1}^{n} w_{ij}z_{ij}$ and minimizing $\sum_{j=1}^{n} y_j$ at the same time, then the objective can also be represented as

$$\text{Maximize } \left(\sum_{i=1}^{p}\sum_{j=1}^{n} w_{ij}z_{ij}\right)/\left(\sum_{j=1}^{n} y_j\right) \qquad (4)$$

The objective in Eq.4 is maximizing the unit weight values for each selected d_j in D'. Eq.4 has the same constraints in Eq.3. Since the final output of the Symp-Med matching problem is a vector D' with maximized unit weight value. An optimal result can be obtained in polynomial time without solving z_{ij} and y_j in Eq.4. A Symp-Med Matching algorithm based on a greedy method is designed to solve this problem.

Alg. 2: Greedy-based Symp-Med Matching
input: weight matrix $W \in R^{p \times n}$, parameter ε'
output: vector D'
initialize:
score vector $A \in R^{1 \times n}$,
index vector $H \in R^{1 \times n}$ stores indexes for elements in D sorted in descending order according to A,
index vector $B \in R^{1 \times p}$
begin
  for $b_i \in B$ **do**
    $b_i = false$
  end for
  for $a_j \in A$ **do**
    $a_j = \sum_{i=1}^{p} w_{ij}$
  end for
  $H \leftarrow sort(A)$
  for $h_j \in H$ **do**
    for $b_i \in B$ **do**
    if $b_i = false$ **and** $w_{ij} > 0$ **then**
      $D' = D' \cup \{d_{h_j}\}$
      $b_i = true$
    end if
    end for
  end for
Return: D'
end.

Alg. 2 applies greedy method. It uses a score vector A to sort $d_j \in D$ in descending order, and an index vector B to indicate if the constraint Eq.2 is satisfied or not. It incrementally extends D' until all the constraints are satisfied.

V. EXPERIMENTS

The motivation of our experiments is two-fold: (1) To examine how the value ε' affect the performance of Branch-and-Cut based Symp-Med Matching Algorithm; (2) To evaluate the performance of Greedy-based Symp-Med Matching algorithm. The rest of this section presents a detailed description of our dataset, experimental design, evaluation methodology, and result analysis.

A. Dataset Description and Evaluation Methodology

We use the clinical notes dataset from the 2009 i2b2 workshop on NLP challenges [16] as experiment dataset. There are 1249 clinical notes in total. After pre-processing, 1239 clinical notes remain. We divided the dataset into 4 groups randomly. Each group has training set and test set. In each group, 155 clinical notes are used as the training set, and 155 clinical notes are used as the test set in each group. We extract about 1215-1346 symptom concepts and 609-664 medication concepts for each training/test set.

We evaluate the accuracy of algorithms using two sets of evaluation metrics: 1) Precision (P) and Recall (R); 2) True Positive Rate (TPR) and False Positive Rate (FPR) [17]. ROC (receiver operating characteristic) curve shows how the true positive rate varies with the false positive rate. The area under the ROC curve (AUC) presents achievable TPR with respect to FPR.

B. Symp-Med Matching Analysis

By varying the value of ε', we obtain average performance results of Branch-and-Cut based Symp-Med Matching from four groups of datasets. The result is shown in Table I.

TABLE I. AVERAGE PERFORMANCE RESULTS OF ALG. 1

ε'	TPR	FPR	Precision	Recall
0.1	0.558	0.080	0.208	0.558
0.2	0.397	0.032	0.311	0.397
0.3	0.313	0.018	0.396	0.313
0.4	0.225	0.009	0.494	0.225
0.5	0.162	0.005	0.553	0.162
0.6	0.133	0.003	0.587	0.133
0.7	0.110	0.003	0.589	0.110
0.8	0.089	0.002	0.582	0.089
0.9	0.068	0.002	0.614	0.068
1.0	0.048	0.001	0.634	0.048

ε' is used to balance the objective of maximizing the total weight value and minimizing the total selected d_j number. $\varepsilon' = 1$ means only adding necessary d_j to result sets, because each time adding a new d_j, it costs the value of $\max_i\left(\sum_{j=1}^{n} w_{ij}\right)$ loss to the total maximum objective function. So when $\varepsilon' = 1$, it achieves the largest precision, but the smallest recall. The average experiment precision is 63.4%, and recall is 4.8%. By decreasing the ε' value, the precision decreases, but the recall increases. When $\varepsilon' = 0.1$, we have the lowest precision, 20.8%, and highest recall, 55.8%. When $\varepsilon' =$

0, the objective of minimizing selected d_j number is not considered. The algorithm returns all the d_j in D which connects to any element in S'. In our experiments, the average precision is 3.74%, and the average recall is 99.7%.

We implement Greedy-based Symp-Med Matching on the four groups of datasets, and the average results are in Table II.

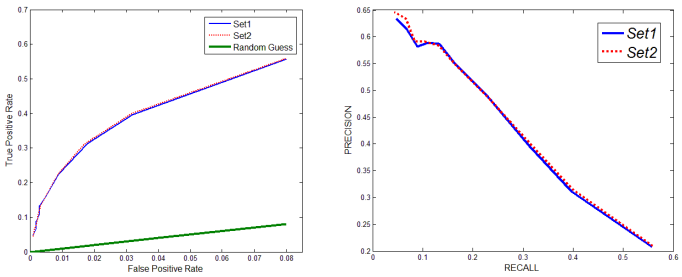
TABLE II. AVERAGE PERFORMANCE RESULTS OF ALG.2

TPR	FPR	P	R
0.061	0.001	0.634	0.061

The objective of Alg.2 is to maximize the unit weight values. The average precision is 63.4%, and the average recall is 6.1%. The result is close to the result in Table I when $\varepsilon' = 1$. The Alg.1 can capture the full spectrum of performances by varying the value of ε', while Alg.2 can produce a good precision result and improve the recall without solving the corresponding LP problem in Alg.1.

We only remove negation concepts by negation annotator and section annotator during pre-processing. There are a lot of noises exist in extracted symptom and medication concepts. Based on the most frequent sections with symptom and medication concepts, we implement our algorithms on the datasets only contain symptom concepts from most frequent sections in clinical notes. Let indicate the experiments on selected sections from clinical notes as Set 2 experiment, and the experiments on all sections as Set 1. The results in Table I and Table II are from Set 1 experiment.

We use ROC curves and Precision-Recall (PR) curve to capture the full spectrum of performances of Set 1 and Set 2 experiments as shown in Fig. 2.



(a) ROC curve (b) PR curve

Fig. 2. Comparison in ROC and PR Curves

As shown in Fig. 2 (a), the ROC curve indicates the set 2 has a better result than set 1, since the AUC is slightly larger in set 2. Both set 2 and set 1 have better performance than the Random Guess result. In Fig. 2 (b), the result indicates set 2 also has a better precision/recall results than set 1. The performances of Symp-Med matching algorithms can be improved if more noises can be removed from extracted symptom and medication concepts in the pre-processing stage.

VI. RELATED WORK

A. Information Extraction from Clinical Notes

How to extract useful information from biomedical text is a long-standing NLP problem. MetaMap is a program developed by NLM (National Library of Medicine) to extract Metathesaurus concepts from texts [8]. It returns different semantic types presented in text. Reference [11] proposes a framework for modeling and mining symptom relationships from clinical notes. Reference [18] proposes a method to identify medical concepts from the SNOMED Clinical Terminology in free texts. There are different types of NLP challenges for clinical narratives [19], such as concept extraction from clinical notes, medical problem concept classification, relation extraction, and so on. MedEx is a medication information extraction system developed for extracting medication names and signatures from clinical narratives [9], it reported a 93.2% F-measure on identifying drug names. Another linguistic approach for identification of medication names and related information in clinical narratives uses negation maker to exclude negation medication information [20].

B. Symptom and Medication Research for Diseases

Clinical symptoms are important for patients to control the exacerbation of diseases [21]. The relationships between symptoms and medication for given one particular disease (such as asthma [22, 23], cancer [24]) have been studied with case study methods and statistical methods. A symptom-medication score is used as an instrument to evaluate the disease severity by recording symptoms and rescue medication [25]. Currently, there is little research work on extracting symptom and medication concepts from clinical notes for medication error detection and surveillance.

C. Bipartite Graph Theory and Application

The bipartite graph is used to represent concepts extracted from texts, and SympGraph uses bipartite to represent symptom information from clinical notes [11]. In this paper, we use a bipartite graph to represent the relationship between symptom concepts and medication concepts extracted from clinical notes. A bipartite graph contains two groups of vertices connected between groups, and no edge among the vertices in the same group. Maximum matching is an important problem for bipartite graph [26]. Our problem in this paper is different from maximum matching problem. Our Symp-Med matching algorithms match at least one edge with positive weight value for a symptom, in the meantime, to maximize the total weight values and minimize the number of medication names. Reference [27] develops neighborhood formation and anomaly detection algorithms for the bipartite graph. The neighborhood formation algorithm is to find similar vertices inside a group, which can be used for symptom expansion and medication expansion.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we present Symp-Med matching framework for representing and mining relationships between the symptom and medication extracted from clinical notes. We formulate the Symp-Med matching problem as an ILP problem, and propose Symp-Med matching algorithms for solving the Symp-Med matching problem. We explore a Branch-and-Cut based Symp-Med matching algorithm to solve the ILP problem, and define a parameter to balance the two objectives in the ILP problem. Then we change the objective function in the ILP problem to a combined maximizing the unit

weight value objective, and propose a Greedy-based Symp-Med matching algorithm for solving it.

Our Symp-Med matching algorithms can be used to predict a set of medications based on a given symptom set. The Symp-Med matching framework can also be applied to error detection[28] for medications in clinical notes. In future work, we plan to improve current work from the following aspects:

1) We build a Symp-Med weight matrix for our Symp-Med framework. We intend to extend to the weight factor set. Currently, we only use the information extracted from experiment clinical notes dataset to build the weight factor set. Only two weight factors are defined in this paper. In the future, we plan to integrate other factors into the weight factor set, such as drug indications, side effects of drugs, drug interactions, drug administration information etc., from publicly available datasets such as DrugBank, RxNorm, and UMLS etc.[29]

2) There are still a lot of noises remained in extracted symptom concepts and medication concepts during clinical notes pre-processing. These noises affect the performance of our Symp-Med matching algorithms. Improving the results of symptom and medication extraction is worthwhile.

3) Currently, we only consider the relationship between symptom concepts and medication concepts. We plan to integrate symptom-symptom and medication-medication relationships into the Symp-Med framework. For example, we plan to use similarity to build a symptom-symptom matrix. This will help to expand and discover more related symptom information for patients based on observed symptoms.

ACKNOWLEDGMENT

This work is supported in part by the NSF grant IIP 1160960 for the center for visual and decision informatics (CVDI).

REFERENCES

[1] Roberts, K. and S.M. Harabagiu, A flexible framework for deriving assertions from electronic medical records. Journal of the American Medical Informatics Association, 2011. 18(5): p. 568-573.

[2] Kim, M.-Y., et al., Patient Information Extraction in Noisy Tele-health Texts, in In the IEEE International Conference on Bioinformatics and Biomedicine (BIBM13)2013: Shanghai, China.

[3] Roque, F.S., et al., Using electronic patient records to discover disease correlations and stratify patient cohorts. PLoS computational biology, 2011. 7(8): p. e1002141.

[4] Hripcsak, G., et al., Mining complex clinical data for patient safety research: a framework for event discovery. Journal of biomedical informatics, 2003. 36(1): p. 120-130.

[5] Pakhomov, S.V., A. Ruggieri, and C.G. Chute. Maximum entropy modeling for mining patient medication status from free text. in Proceedings of the AMIA Symposium. 2002. American Medical Informatics Association.

[6] Henriksson, A., Semantic Spaces of Clinical Text: Leveraging Distributional Semantics for Natural Language Processing of Electronic Health Records. 2013.

[7] Chapman, W.W., et al., Overcoming barriers to NLP for clinical text: the role of shared tasks and the need for additional creative solutions. Journal of the American Medical Informatics Association, 2011. 18(5): p. 540-543.

[8] Aronson, A.R. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. in Proceedings of the AMIA Symposium. 2001. American Medical Informatics Association.

[9] Xu, H., et al., MedEx: a medication information extraction system for clinical narratives. Journal of the American Medical Informatics Association, 2010. 17(1): p. 19-24.

[10] Aronson, A.R., Metamap: Mapping text to the umls metathesaurus. Bethesda, MD: NLM, NIH, DHHS, 2006.

[11] Sondhi, P., et al. SympGraph: a framework for mining clinical notes through symptom relation graphs. in Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. 2012. ACM.

[12] Williamson, D.P., The primal-dual method for approximation algorithms. Mathematical Programming, 2002. 91(3): p. 447-478.

[13] Mitchell, J.E., Branch-and-cut algorithms for combinatorial optimization problems. Handbook of Applied Optimization, 2002: p. 65-77.

[14] Makhorin, A., GNU linear programming kit. Moscow Aviation Institute, Moscow, Russia, 2001. 38.

[15] RinnooyKan, A. and J. Telgen, The complexity of linear programming. Statistica Neerlandica, 1981. 35(2): p. 91-107.

[16] Uzuner, Ö., I. Solti, and E. Cadag, Extracting medication information from clinical text. Journal of the American Medical Informatics Association, 2010. 17(5): p. 514-518.

[17] Davis, J. and M. Goadrich. The relationship between Precision-Recall and ROC curves. in Proceedings of the 23rd international conference on Machine learning. 2006. ACM.

[18] Patrick, J., Y. Wang, and P. Budd. An automated system for conversion of clinical notes into SNOMED clinical terminology. in Proceedings of the fifth Australasian symposium on ACSW frontiers-Volume 68. 2007. Australian Computer Society, Inc.

[19] Uzuner, Ö., et al., 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. Journal of the American Medical Informatics Association, 2011. 18(5): p. 552-556.

[20] Hamon, T. and N. Grabar, Linguistic approach for identification of medication names and related information in clinical narratives. Journal of the American Medical Informatics Association, 2010. 17(5): p. 549-554.

[21] Bruch, H. and I. Hewlett, Clinical Notes Psychologic Aspects of the Medical Management of Diabetes in Children. Psychosomatic Medicine, 1947. 9(3): p. 205-209.

[22] Main, J., et al., The use of reliever medication in asthma: the role of negative mood and symptom reports. Journal of Asthma, 2003. 40(4): p. 357-365.

[23] Slaughter, J.C., et al., Effects of ambient air pollution on symptom severity and medication use in children with asthma. Annals of Allergy, Asthma & Immunology, 2003. 91(4): p. 346-353.

[24] Riechelmann, R.P., et al., Symptom and medication profiles among cancer patients attending a palliative care clinic. Supportive Care in Cancer, 2007. 15(12): p. 1407-1412.

[25] Häfner, D., et al., Prospective validation of 'Allergy-Control-SCORETM': a novel symptom–medication score for clinical trials. Allergy, 2011. 66(5): p. 629-636.

[26] Hopcroft, J.E. and R.M. Karp, An n^5/2 algorithm for maximum matchings in bipartite graphs. SIAM Journal on computing, 1973. 2(4): p. 225-231.

[27] Sun, J., et al. Neighborhood formation and anomaly detection in bipartite graphs. in Data Mining, Fifth IEEE International Conference on. 2005. IEEE.

[28] Ling, Y., et al., An Error Detecting and Tagging Framework for Reducing Data Entry Errors in Electronic Medical Records (EMR) System, in In the IEEE International Conference on Bioinformatics and Biomedicine (BIBM13)2013: Shanghai, China.

[29] Samwald, M., et al., Linked open drug data for pharmaceutical research and development. Journal of cheminformatics, 2011. 3(1): p. 19.