[Query algorithms for unstructured data in MIMIC II are performed using NLP processing techniques. [Perl]

#use strict;

use Data::Dumper;

#open drug file output

open (DRUGFILE, ">nlp\_found\_admission\_drugs.txt") || die "output nlp\_found\_admission\_drugs.txt $!";

#open nlp log file

open (LOGFILE, ">nlp.log") || die "output nlp.log $!";

#open file output for pt that had transfer meds

open (TFILE, "> TRANSFER\_SID.list") || die "output TRANSFER\_SID.list $!";

#open pt cohort file output (list of patients with home/admission medication in their discharge summaries)

open (ALL\_PT\_WITH\_HOME\_MEDS\_FILE, ">ALL\_PT\_WITH\_HOME\_MEDS\_HEADER.list") || die "output ALL\_PT\_WITH\_HOME\_MEDS\_HEADER.list $!";

$drug\_list\_fname = $ARGV[0]; #list of drugs we are interested in

$ds\_fname = $ARGV[1]; #discharge summary txt file

open DFILE, $drug\_list\_fname or die "Cannot open $drug\_list\_fname";

open FILE, "$ds\_fname" || die "Cant open $ds\_fname $!"; #discharge summaries

@drugs=<DFILE>;

%drugs=();

&load\_drugs();

&find\_drugs();

#print @drugs;

close DRUGFILE;

close TFILE;

close ALL\_PT\_WITH\_HOME\_MEDS\_FILE;

#===================================================

# load the list of drugs that we are looking for...

sub load\_drugs {

while ($line = <DFILE>) {

#print $line;

$line = &remcr($line);

if (length($line) > 1) {

# $line =~ tr/[a-z]/[A-Z]/; # Convert the line to upper case.

$line = lc $line;

#PPI:

if ( $line =~ /omeprazole|prilosec|esomeprazole|nexium|pantoprazole|protonix|lansoprazole|prevacid|rabeprazole|aciphex/i ) {

# H2B:

if ( $line =~ /ranitidine|zantac|famotidine|pepcid|cimetidine|tagamet|axid|nizatidine/i ) {

# DIURETICS:

if ( $line =~ /furosemide|lasix|hydrochlorothiazide|hctz|spironolactone|aldactone|torsemide|demadex|acetazolamide|diamox|nizatidine|triamterene|dyrenium|bumetanide|bumex|ethacrynic|edecrin|eplerenone|inspra|amiloride|midamor|metolazone|mykrox|zaroxolyn|chlorthalidone|hygroton|thalitone/i ) {

$drugs{$line} = 0b0010;

} else { #should never happen

print "drugs list does not match what's in perl code '$line' \n";

print "$line \n";

die;

}

}

}

close DFILE;

}

sub remcr {

my ($line) = @\_;

while ($line =~ /[\n\r]$/){chop($line);}

return ($line);

}

sub find\_drugs {

my $meds = ""; # Contains the section header which indicates that we're still in a medications section

my $sect = "";

my $type = "";

my $ty = 0;

my $medgroup = 0;

my $found = 0;

my $group = 0;

my $admit = 'unk';

my $disch = 'unk';

my $other = 'unk';

my @words = ();

my $sect\_head\_line\_index = 0;

my $line\_count = 0;

$admit = 'unk'; $disch = 'unk'; $other = 'unk'; $ty = 0;

my $prev\_line\_blank = 0;

my $found\_home\_meds = 0;

while ($line = <FILE>) {

if ($line =~ /(\d+)\_:-:\_(\d+)\_:-:\_/) {

$hadmid = $1; $case = $2;

#print "$hadmid\n";

$found\_home\_meds = 0;next;

}

chomp($line);

$line\_count++;

#$line = &remcr($line);

#$line =~ tr/[a-z]/[A-Z]/; # Convert the line to upper case. commented out on 12/18/11

## section head in ds

#if ($line =~ /^((\d|[A-Z])(\.|\)))?\s\*([a-zA-Z][a-zA-Z',\.\-\\*\d\[\] ]+)(:|;|WERE|IS|INCLUDED|INCLUDING)/)

#if ($line =~ /^((\d|[A-Z])(\.|\)))?\s\*([a-zA-Z',\.\-\\*\d\[\]\(\) ]+)(:|;|WERE| IS |INCLUDED|INCLUDING)/) {

#if ($prev\_line\_blank & ($line =~ /^((\d|[A-Z])(\.|\)))?\s\*([a-zA-Z',\.\-\\*\d\[\]\(\) ]+)(:|;|WERE| IS |INCLUDED|INCLUDING|were| is | included|including)/)) {

if (($prev\_line\_blank && ($line =~ /^((\d|[A-Z])(\.|\)))?\s\*([a-zA-Z',\.\-\\*\d\[\]\(\) ]+)(:|;|WERE| IS |INCLUDED|INCLUDING|were| is | included|including)/)) || ($line =~ /^((\d|[A-Z])(\.|\)))?\s\*(A-Z[a-zA-Z',\.\-\\*\d\[\]\(\) ]+)(:|;|WERE| IS |INCLUDED|INCLUDING|were| is | included|including)/)) {

print LOGFILE "$case potential section heading:$line\n";

$sect = $4;

#print "$sect\n";

if ($meds) { ## med section ended, now in non-meds section

#if this section header starts with meds or medications and it's immediately below another header,

#then treat this as part of the previous section

# this is for catching the following types of scenarios:

#Medications on Admission:

#Meds: Furosemide 10mg qday, metoprolol 12.5mg bid, MVI,

if ($sect =~ /^[^a-zA-Z]\*med(ication)?(s)?/i && ($line\_count == $sect\_head\_line\_index+1) && ($type eq 'admission') && ($sect !~ /discharge|transfer/i) ) {

#treat this as part of the previous meds section

print "Treat As Same Section $sect\n";

} else { #this is start of a new section

$meds = "";

$type= "";

print LOGFILE "$case meds section ended:$line\n";

}

}

$sect\_head\_line\_index = $line\_count;

# print "---->$3\n";

#if $type is "" (i.e. $type is not already set) and the section header contains medications or meds

if ( !$type && $sect =~ /medication|meds/i) { ## new meds section of some type

print LOGFILE "$case meds section started:$sect\n";

$meds = $sect;

$found = 0;

#first criteria does pattern matching on $line (instead of just on $meds)

#IF previous line is blank and this line starts with something like Meds: or Medications: (potentially followed by some other words on the same line

#or IF this line consists of just Meds or Medications or Meds: Medications: or Medication: (and nothing else following it), then we declare this as a HOME medication section

if ($prev\_line\_blank && ($line=~ /\A\s\*(\d)\*.?\s\*med(ication)?s?:\s\*/i) ||

$line =~ /\A\s\*(\d)\*.?\s\*med(ication)?s?:?\s\*\Z/i) {

$type = 'admission'; $ty = 1;

}

elsif ($meds =~ /admission|admitting/i){$type = 'admission'; $ty = 1;}

elsif ($meds =~ /presentation|baseline/i){$type = 'admission'; $ty = 1;}

elsif ($meds =~ /home|nh|nmeds/i){$type = 'admission'; $ty = 1;}

elsif ($meds =~ /pre(\-|\s)?(hosp|op)/i){$type = 'admission'; $ty = 1;}

elsif ($meds =~ /current|previous|outpatient|outpt|outside/i){$type = 'admission'; $ty = 1;}

#elsif ($meds =~ /^[^a-zA-Z]\*med(ication)?(s)?/i){$type = 'admission'; $ty = 1;}

elsif ($meds =~ /^Maternal/i){$type = 'admission'; $ty = 1;}

elsif ($meds =~ /transfer|xfer/i){$type = 'transfer'; $ty = 4;} #we don't want transfer meds LL, 12/13/11

elsif ($meds =~ /discharge/i){$type = 'discharge'; $ty = 2;}

else{$type = $meds; $ty = 3;} ## type other

if (($ty == 1) && ($admit eq 'unk')){$admit = 'no';} ## unk -> no -> yes

elsif (($ty == 2) && ($disch eq 'unk')){$disch = 'no';}

elsif (($ty == 3) && ($other eq 'unk')){$other = 'no';}

if ($type eq 'admission' && !$found\_home\_meds) {

print ALL\_PT\_WITH\_HOME\_MEDS\_FILE "$case $hadmid\n"; #output the subject id of all patients

$found\_home\_meds = 1;

}

if ($ty == 4) {

print TFILE "$case\n";

}

} #end if section is medication|meds

#} elsif ($line =~ /medication|meds/i && $line =~ /admission|discharge|transfer/i) {

} elsif ($line =~ /medication|meds/i && $line =~ /admission/i) { #else if this is not a section header but contains the words admission/medications, output to the log file

print LOGFILE "$case matches admission medication|meds, but not section heading:$sect\n";

}

if ($meds) { ## in meds section, look at line

#@words = split (/[- ,\.\d\)]+/,$line);

# @words = split (/[ ,\.\d\)\_\W\s]+/ ,$line); #LL 12/13/11 added \W \s as a separator

# foreach $word (@words) {

# $word =~ tr/[a-z]/[A-Z]/;

$line = lc $line;

@words = split(/[,\.]+/, $line);#SC can't split on whitespace b/c some drugs 2+ words

foreach $word (@words){#SC split on , and . to ensure that lines with multiple drugs catch them all--could create a problem if

#unpunctuated lists occur

#PPI:

if ($word =~ /omeprazole|prilosec|esomeprazole|nexium|pantoprazole|protonix|lansoprazole|prevacid|rabeprazole|aciphex/i){#PPI

#H2B:

if ($word =~ /ranitidine|zantac|famotidine|pepcid|cimetidine|tagamet|axid|nizatidine/i){#H2B

#DIURETICS:

if ($word =~ /furosemide|lasix|hydrochlorothiazide|hctz|spironolactone|aldactone|torsemide|demadex|acetazolamide|diamox|nizatidine|triamterene|dyrenium|bumetanide|bumex|ethacrynic|edecrin|eplerenone|inspra|amiloride|midamor|metolazone|mykrox|zaroxolyn|chlorthalidone|hygroton|thalitone/i){#DIURETICS

# my start = length($`);

# my $str = substr $line, $start;

# $word = lc $word;

# print "$word\n";

# if ($drugs{$word}) {

if ($type eq 'admission') { #only print if it is home/admission meds

#print DRUGFILE "$case|$type|$word\n";

# print DRUGFILE "$case\t$hadmid\t$word\n";

print DRUGFILE "$case,$hadmid,$&,\n"#$word,\n"

}

#Add to the meds group if you haven't already

#SC $medgroup = $medgroup | $drugs{$word};

if ($ty == 1){$admit = 'yes';}

elsif ($ty == 2){$disch = 'yes';}

elsif ($ty == 3){$other = 'yes';}

#print OFILE "$case|$type|$line\n";

$found = 1;

}

}

} #end of if ($meds)

#check if this is a blank line

if ($line =~ /^$/ || length(chomp($line))==0 || $line !~ /[a-zA-Z\d]/) {

$prev\_line\_blank = 1;

#print "this is blank $line\n";

} else {

$prev\_line\_blank = 0;

#$len = length($line);

#print "NOT BLANK: $line\n";

}

} #END while each line

} #END SUB