medExtractR_tapering Vignette

2022-06-06

Introduction

The `medExtractR` package uses a natural language processing (NLP) system called *medExtractR*.[1] This system is a medication extraction system that uses regular expressions and rule-based approaches to identify key dosing information including drug name, strength, dose amount, frequency or intake time, dose change, and last dose time. In this vignette, we introduce the `medExtractR_tapering` function, an extension on the basic `medExtractR` function. The goal of this function is to extract more information for drugs following a tapering schedule. Some drugs (e.g., steroids, antidepressants) require weaning schedules to be gradually stopped in order to avoid adverse side effects or withdrawal symptoms. Tapering schedules are much more complex and require more entities in order to build longitudinal dose data.

Tapering extension to `medExtractR`

Similar to `medExtractR`'s `extract_entities` subfunction, the tapering extension has its own corresponding subfunction called `extract_entities_tapering`. In addition to the entities identified by the basic `medExtractR` function, the tapering extension can also identify: dispense amount (entity name in output: "DispenseAmt"); number of refills ("Refill"); keywords indicating a particular dose schedule pattern, such as "taper", "wean", "dose pack", or "discontinue" ("DoseSchedule"); keywords indicating whether a dose is past, current, or future ("TimeKeyword"); words or symbols linking consecutive doses in a tapering schedule ("Transition"); and prepositions adjacent to other entities ("Preposition"), used in some contexts to preserve or infer meaning.

```r
library(medExtractR)

# prednisone tapering note file name
pred_fn <- c(
  system.file("examples", "predpid1_2013-04-27_note1_1.txt", package = "medExtractR"),
  system.file("examples", "predpid2_2015-12-11_note1_1.txt", package = "medExtractR"),
  system.file("examples", "predpid3_2017-01-01_note1_1.txt", package = "medExtractR")
)

# execute medExtractR_tapering
pred_mxr <- do.call(rbind, lapply(pred_fn, function(filename){
  pred_note <- paste(scan(filename, '', sep = '\n', quiet = TRUE), collapse = '\n')
  fn <- sub(".+/", "", filename)
  cbind("filename" = fn,
        medExtractR_tapering(note = pred_note,
                             drug_names = c("prednisone", "pred"),
                             unit = "mg"))
}))
```

This extended function was created to handle three common patterns of dosing found in notes for patients being treated with medications that commonly use a tapering schedule to gradually decrease the patient's dose. The drug prednisone is an example of one such medication, and de-identified notes for patients being treated with prednisone were used during `medExtractR_tapering` development (described in a later section).

We call these additional dosing patterns “taper I”, “taper II”, and “alternating” schedules. In the subsections below, we explain the structure of each of these schedules and present an example of `medExtractR_tapering` being applied to each.

Taper I

Taper I schedules are tapering regimens in which consecutive doses are explicitly written out. Examples of taper I schedules include:

- “prednisone 10 mg tablet 4 tablets by mouth daily for 2 weeks then 3 tablets by mouth daily for 1 week then 2 tablets by mouth daily for 1 week then 1.5 tablets by mouth daily for 1 week then 1 tablet by mouth daily for 1 week then 0.5 tablet by mouth daily for 1 week”
- “will increase prednisone to 15 mg once a day for 5-7 days, then taper (10 mg to 5 mg to off) over 2 weeks”
- “prednisone 5 mg daily for 7 days and 2.5 mg daily for another 3 days”

In a taper I schedule, at least some dose information is available for two or more segments of a weaning schedule. In the first example above, the strength of an individual tablet is provided up front, and each segment of the taper schedule just provides the number of tablets to take. This represents an ideal case where a clear duration is available for each segment. In the second example, the durations associated with each segment are less clear. MedExtractR only identifies the information exactly as it’s written in a note, but a post-processing function could determine how to assign daily dosages given this information (e.g., 15 mg for 6 days, 10 mg for 7 days, 5 mg for 7 days, 0 mg).

```r
# Taper 1 schedule
subset(pred_mxr, filename == sub(".+/", "", pred_fn[1]))
```

```
##                           filename      entity       expr     pos
## 1  predpid1_2013-04-27_note1_1.txt    DrugName prednisone 561:571
## 2  predpid1_2013-04-27_note1_1.txt    Strength      10 mg 572:577
## 3  predpid1_2013-04-27_note1_1.txt     DoseAmt          4 585:586
## 4  predpid1_2013-04-27_note1_1.txt       Route   by mouth 595:603
## 5  predpid1_2013-04-27_note1_1.txt   Frequency      daily 604:609
## 6  predpid1_2013-04-27_note1_1.txt Preposition        for 610:613
## 7  predpid1_2013-04-27_note1_1.txt    Duration    2 weeks 614:621
## 8  predpid1_2013-04-27_note1_1.txt  Transition       then 622:626
## 9  predpid1_2013-04-27_note1_1.txt     DoseAmt          3 627:628
## 10 predpid1_2013-04-27_note1_1.txt       Route   by mouth 637:645
## 11 predpid1_2013-04-27_note1_1.txt   Frequency      daily 646:651
## 12 predpid1_2013-04-27_note1_1.txt Preposition        for 652:655
## 13 predpid1_2013-04-27_note1_1.txt    Duration     1 week 656:662
## 14 predpid1_2013-04-27_note1_1.txt  Transition       then 663:667
## 15 predpid1_2013-04-27_note1_1.txt     DoseAmt          2 668:669
## 16 predpid1_2013-04-27_note1_1.txt       Route   by mouth 678:686
## 17 predpid1_2013-04-27_note1_1.txt   Frequency      daily 687:692
## 18 predpid1_2013-04-27_note1_1.txt Preposition        for 693:696
## 19 predpid1_2013-04-27_note1_1.txt    Duration     1 week 697:703
## 20 predpid1_2013-04-27_note1_1.txt  Transition       then 704:708
## 21 predpid1_2013-04-27_note1_1.txt     DoseAmt        1.5 709:712
## 22 predpid1_2013-04-27_note1_1.txt       Route   by mouth 721:729
## 23 predpid1_2013-04-27_note1_1.txt   Frequency      daily 730:735
## 24 predpid1_2013-04-27_note1_1.txt Preposition        for 736:739
## 25 predpid1_2013-04-27_note1_1.txt    Duration     1 week 740:746
## 26 predpid1_2013-04-27_note1_1.txt  Transition       then 747:751
## 27 predpid1_2013-04-27_note1_1.txt     DoseAmt          1 752:753
## 28 predpid1_2013-04-27_note1_1.txt       Route   by mouth 761:769
## 29 predpid1_2013-04-27_note1_1.txt   Frequency    daily 770:775
## 30 predpid1_2013-04-27_note1_1.txt Preposition      for 776:779
## 31 predpid1_2013-04-27_note1_1.txt    Duration   1 week 780:786
## 32 predpid1_2013-04-27_note1_1.txt  Transition     then 787:791
## 33 predpid1_2013-04-27_note1_1.txt     DoseAmt      0.5 792:795
## 34 predpid1_2013-04-27_note1_1.txt       Route by mouth 803:811
## 35 predpid1_2013-04-27_note1_1.txt   Frequency    daily 812:817
## 36 predpid1_2013-04-27_note1_1.txt Preposition      for 818:821
## 37 predpid1_2013-04-27_note1_1.txt    Duration   1 week 822:828
```

Taper II

Taper II schedules are tapering regimens in which a repetitive dosing change is written more concisely. Typically, this is written in the form of *“ change by at ”*. As an example, consider the phrase *“Take prednisone 5mg tablet 4 tabs daily for one week, then decrease by 5mg every week then stop”* In this excerpt, the patient’s initial dose is 20 mg per day for a duration of one week. From the second part of the phrase, we can infer that after the first week, the patient’s dosage will be 15 mg daily for the second week, 10 mg daily for the third week, 5 mg daily for the fourth week, and then 0 mg (off) after that.

```
# Taper 2 schedule
subset(pred_mxr, filename == sub(".+/", "", pred_fn[2]))
```

```
##                            filename       entity          expr     pos
## 38 predpid2_2015-12-11_note1_1.txt     DrugName    prednisone 474:484
## 39 predpid2_2015-12-11_note1_1.txt     Strength          5 mg 485:489
## 40 predpid2_2015-12-11_note1_1.txt DoseStrength          20mg 502:506
## 41 predpid2_2015-12-11_note1_1.txt   IntakeTime every morning 507:520
## 42 predpid2_2015-12-11_note1_1.txt  Preposition           for 521:524
## 43 predpid2_2015-12-11_note1_1.txt     Duration       14 days 525:532
## 44 predpid2_2015-12-11_note1_1.txt   Transition          then 533:537
## 45 predpid2_2015-12-11_note1_1.txt   DoseChange      decrease 538:546
## 46 predpid2_2015-12-11_note1_1.txt  Preposition            by 547:549
## 47 predpid2_2015-12-11_note1_1.txt DoseStrength          5 mg 550:554
## 48 predpid2_2015-12-11_note1_1.txt     Duration  every 7 days 555:567
## 49 predpid2_2015-12-11_note1_1.txt  Preposition         until 568:573
## 50 predpid2_2015-12-11_note1_1.txt DoseSchedule          done 574:578
```

Alternating

Alternating schedules consist of a patient being on multiple doses (typically two), with different dosages on consecutive days. The patient may have two specific doses, for example *“20mg of prednisone on even days and 10mg of prednisone on odd days”*, or they may be on a single dose but taking it intermittently, for example *“take prednisone 5mg tablet every other day for one week”*.

```
# Alternating schedule
subset(pred_mxr, filename == sub(".+/", "", pred_fn[3]))
```

```
##                            filename       entity            expr     pos
## 51 predpid3_2017-01-01_note1_1.txt     DrugName            pred 630:634
## 52 predpid3_2017-01-01_note1_1.txt DoseStrength          2.5 mg 635:641
## 53 predpid3_2017-01-01_note1_1.txt        Route          orally 642:648
## 54 predpid3_2017-01-01_note1_1.txt    Frequency every other day 649:664
## 55 predpid3_2017-01-01_note1_1.txt DoseSchedule       alternate 665:674
## 56 predpid3_2017-01-01_note1_1.txt  Preposition            with 675:679
## 57 predpid3_2017-01-01_note1_1.txt DoseStrength            5 mg 680:684
## 58 predpid3_2017-01-01_note1_1.txt    Frequency every other day 685:700
```

Note: It is possible for a dosing regimen to be written in a mixture of these cases. For example, the phrase *"restart pred at 40 mg daily for 1 week, 30 mg daily for one week, 20 mg daily for one week and then drop by 5 mg per week after that"* starts as a taper I schedule, but ends as a taper II schedule. An individual segment of a taper I schedule could also involve an alternating dose, particularly taking a single dose every other day in weaning schedules.

`medExtractR_tapering` development

The tapering extension to medExtractR was developed on a set of 162 training notes from Vanderbilt University's de-identified EHR system, the Synthetic Derivative (SD). As a first step, we developed annotation guidelines as a way to rigorously define the definition of each new entity (or changes to any existing entity, if applicable). Two reviewers would then independently annotate each note, and a third party group would help resolve any discrepancies. The annotation guidelines were then updated, and annotations were revised, based on this process.

This set of training notes was also used to generate the default entity dictionaries included within the `medExtractR` package. The annotations, which serve as the gold standard for what information should be extracted, were used as the starting point for all dictionaries. Those expressions were then removed (e.g., mispelled words) or generalized using regular expressions (e.g., changing a "2 days" duration to "\d{1,2} days", where \d represents any one or two digit number). Examples of some expressions in the new entity dictionaries are provided below:

- Duration: much of the duration dictionary is written using regular expressions to account for various numbers being used. Examples of expressions that would be considered durations are: `2 weeks`, `next 3 days`, `1 week ago`, `weekly`, `day five`, and dates (for example, to capture expressions like *"Remain on prednisone 5mg daily until April 15"*).
- DoseSchedule: words indicating a specific dosing schedule (`taper`, `weaning`, `dose pack`, etc.) or stopping keywords (`off`, `stop`, `discontinue`, etc.)
- TimeKeyword: `was taking`, `yesterday`, `remain`, `tomorrow`, `current`, etc.
- Transition: `then`, `followed by`, and punctuation such as `;` or `,`.
- Preposition: `from`, `of`, `by`, `on`, `to`, `at`, etc. If a preposition is part of an expression for another entity (e.g., "by" in "by mouth"), then it is not extracted separately as a preposition.

Before running `medExtractR_tapering`, we first pre-processed the notes by censoring any datetime expressions. The Duration entity can include date expressions, however it was common for the SD notes to have de-identified datetime expressions unrelated to dosing. Instead, these expressions typically reflected when the note was created, the time of the appointment, or when a prescription was faxed to the pharmacy. We also removed any notes that used an XML format, due to the structure of these notes being vastly different from the typical free-text clinical notes. XML, or "extensive markup language," is a slightly more structured data format that includes tags to describe pieces of text within the note. The presence of such tags especially interfered with rule-based entities. For example, a dose amount of 3 tablets may appear as `"<DOSE_AMOUNT>3</DOSE_AMOUNT><DOSE_UNIT>tablets</DOSE_UNIT>"`. The `medExtractR` system would typically consider a number followed by the word "tablets" to be a dose amount, but here the tag labels block these two expressions. Two XML-based notes were removed, resulting in 160 notes used for development.

Details of `medExtractR_tapering` functionality

The tapering extension of the `medExtractR` function is capable of extracting more entities associated with tapering schedules, and is thus more flexible than the basic version of `medExtractR`. The arguments between the basic and extended versions of `medExtractR` are the same with one major exception: `medExtractR_tapering` does *not* have an argument for `window_length`. This is because the length of the window in which all relevant entities are contained varies greatly depending on the type of dosing schedule (e.g., taper I, taper II, or alternating). Unlike the basic `medExtractR` function, which uses a fixed window length, `medExtractR_tapering` dynamically generates a search window. This is done by first allowing a very generous search window, and then truncating based on two criteria: (1) the presence of competing drug names or terms; and (2) the distance between consecutive entities. The user can modify this search window process by adding or removing terms to the `drug_list` argument, for example including additional categories of competing terms to use for truncation (e.g., symptoms, procedure, or laboratory names).

```
note <- paste(scan(filename, '', sep = '\n', quiet = TRUE), collapse = '\n')
medExtractR_tapering(note, drug_names, unit, max_dist, ...)
```

The remainder of this section outlines how `medExtractR_tapering` processes a note, from free-text input to a structured entity-level output. We'll use the following excerpt as our example input to help illustrate how the output changes at each step.

```
Chief complaint/Reason for Visit: 4mo follow up
History of Present Illness: No muscle pain since prior visit. Does report increased fatigue.
Also reports R hand tremor intermittently.
Medications:
- Lialda 1.2 gram tablet,delayed release 4 tablets by mouth daily
- prednisone 5mg 2 tablets by mouth daily
- omeprazole 20 mg one tablet by mouth daily
Allergies:
- Penicillins (severe hives)
Past Medical and Surgical History: I have reviewed the problem list and there is no change
since last visit.
Social History:
Exercise - no change since last visit
Tobacco - no smoking
Ethanol - 1 drink per week
Employment - no change since last visit
Impression & Plan:
Doing very well today w/o symptoms and good muscle strength on exam.
1. Check CBC, CMP, CK today, and will have pt check CBC locally in 3mo
2. Taper prednisone: 10mg daily for 7 days then 5mg daily for 7 days then decrease by 1mg
every 7 days until done. Patient is on track with schedule.
3. Continue azathioprine at 200mg daily 4. Can continue to hold alendronate Follow up in
6 months.
```

Step 1: Initially, the search window is defined from the end of the previous drug mention (or the start of the note) up until the start of the next drug mention (or the end of the note). From there, `medExtractR_tapering` relies heavily upon expressions within the `drug_list` argument to further shorten the search window, truncating at the closest match on either side of the drug name of interest. The default `drug_list` is based on cleaned and formatted extractions from RxNorm[3] just like the basic version of `medExtractR`. This search window is what `medExtractR_tapering` provides to the `extract_entities_tapering` function for each drug mention.

Example: In our example note, there are two mentions of prednisone. For mention #1, the drug names "Lialda" (before) and "omeprazole" (after) serve as the boundaries of the search window. For mention #2, "Ethanol" (before) and "azathioprine" (after) are the boundaries. The search window for each mention would be:

```
MENTION 1:
1.2 gram tablet,delayed release 4 tablets by mouth daily
- prednisone 5mg 2 tablets by mouth daily
-

MENTION 2:
 - 1 drink per week
```

```
Employment - no change since last visit
Impression & Plan:
Doing very well today w/o symptoms and good muscle strength on exam.
1. Check CBC, CMP, CK today, and will have pt check CBC locally in 3mo
2. Taper prednisone: 10mg daily for 7 days then 5mg daily for 7 days then decrease by
1mg every 7 days until done. Patient is on track with schedule.
3. Continue
```

Note that these are assuming `drug_list` incorporates only drug names, which is the default. If the user chose to augment this list with symptoms, procedures, or laboratory names, the starting point window for mention #2 could be shortened to just after "CBC" (complete blood count).

Step 2: The `extract_entities_tapering` function extracts all entities found within the initial search window. Strength, dose amount, dose strength, dispense amount, and refill are rule-based entities; all others are primarily dictionary-based. In the basic version of `medExtractR`, if a strength-like expression (e.g., "10 mg") appears in a search window with no dose amount, we would classify that as dose strength. However, in tapering schedules, it is common to have a mix of information within the same phrase, e.g., *"prednisone 10mg tablet 20mg (2 tabs) daily for 2 weeks then 10mg (1 tab) daily for 2 weeks"*. For this reason, `medExtractR` relies more on the context of each strength expression to determine whether it is a strength or a dose strength. For example, if the strength expression is followed by the word "tablet" (or capsule, pill, etc.), we consider it to be "Strength" (i.e., the amount of drug per dosage unit).

Example: Below, we see that `medExtractR` initially extracts all entities it identifies within the search windows above. Note that for mention #1, "1.2 gram" is not extracted as a strength because for prednisone, we set `unit = "mg"`, and hence "gram" is not a relevant unit.

```
MENTION 1:
   entity        expr     pos
  DoseAmt           4 250:251
    Route    by mouth 252:260
Frequency       daily 261:266
 DrugName  prednisone 269:279
 Strength         5mg 280:283
  DoseAmt           2 284:285
    Route    by mouth 294:302
Frequency       daily 303:308


MENTION 2:
      entity       expr     pos
    Duration   per week 603:611
DoseSchedule      Taper 818:823
    DrugName prednisone 824:834
DoseStrength       10mg 836:840
   Frequency      daily 841:846
 Preposition        for 847:850
    Duration     7 days 851:857
  Transition       then 858:862
DoseStrength        5mg 863:866
   Frequency      daily 867:872
 Preposition        for 873:876
    Duration     7 days 877:883
  Transition       then 884:888
  DoseChange   decrease 889:897
 Preposition         by 898:900
DoseStrength        1mg 901:904
```

```
    Duration every 7 days 905:917
 Preposition        until 918:923
DoseSchedule         done 924:928
 Preposition           on 941:943
 Preposition         with 950:954
 TimeKeyword     Continue 970:978
```

Step 3: After extracting all entities but before returning results, `extract_entities_tapering` attempts to remove entities that are not likely to be associated with the drug name of interest. This is done by considering the gap, in number of characters, between consecutive entities both before and after the drug mention (the transition, preposition, and time keyword entities are excluded from this procedure due to the high potential for false positives). If this gap exceeds 50 characters, we remove all extracted entities at and further from the drug mention. The value of 50 characters was selected based on gold standard annotations on a training set of notes. Finally, we remove any prepositions or transitions from the very start or very end of the extractions, as these are primarily linkage entities and would require information on either side.

Example: There are no changes to mention #1 for this step. For mention #2, the first duration extraction of "per week" (`603:611`) has a gap of 207 characters until the next extraction of "Taper" (`818:823`). Since this exceeds the gap size of 50 characters, "per week" is removed as an unrelated entity.

```
MENTION 1:
   entity       expr     pos
 DrugName prednisone 269:279
 Strength        5mg 280:283
  DoseAmt          2 284:285
    Route   by mouth 294:302
Frequency      daily 303:308



MENTION 2:
      entity         expr     pos
DoseSchedule        Taper 818:823
    DrugName   prednisone 824:834
DoseStrength         10mg 836:840
   Frequency        daily 841:846
 Preposition          for 847:850
    Duration       7 days 851:857
  Transition         then 858:862
DoseStrength          5mg 863:866
   Frequency        daily 867:872
 Preposition          for 873:876
    Duration       7 days 877:883
  Transition         then 884:888
  DoseChange     decrease 889:897
 Preposition           by 898:900
DoseStrength          1mg 901:904
    Duration every 7 days 905:917
 Preposition        until 918:923
DoseSchedule         done 924:928
 Preposition           on 941:943
 Preposition         with 950:954
 TimeKeyword     Continue 970:978
```

Step 4: With the results returned from `extract_entities_tapering`, `medExtractR_tapering` performs some additional processing to reduce false positives. First, we look for the first occurrence of punctuation before the drug name (e.g., a period or line break), and remove all entities before that punctuation. For example, in the phrase *“…Patient was previously taking omeprazole 20mg per day. Currently on prednisone 10mg daily…”*, the information “20mg” and “per day” would initially be extracted, since the window wouldn’t truncate until the end of the competing drug name “omeprazole.” However, since there is a period between the two sentences, `medExtractR_tapering` would assume “20mg” and “per day” were not related to prednisone, and thus remove them from the results. The one exception is if the adjacent entity is a preposition, which indicates it may be linking the drug mention of interest to other dosing information (e.g., *“20mg of\nprednisone”*, where “\n” is a line break). If the search window was truncated due to another drug name, we check to see if any of the entities we extracted are immediately adjacent (within 2 characters) to the competing drug name. If so, we continue to remove immediately adjacent entities until a gap of more than two characters occur (ignoring any mentions of “tablet” next to dose amount or strength), at which point we no longer attribute those extractions to the competing drug name.

Example: For mention #1 in our example, when we look at excerpt of the note between prednisone and the closest preceding entity, we see “daily\n- prednisone”, where “\n” is a line break. Since there is punctuation between these two entities (in this case, either \n or - would be sufficient even without the other), we remove the extraction “daily” as well as all other extractions before it.

For mention #2, the search window was shortened by the presence of the drug name “azathioprine”, and `medExtractR` stores the position of this competing drug name (`979:991`). Since “Continue” was extracted immediately before the competing drug name (`970:978`), we remove it from our results. No other entities are removed since there’s a gap between “Continue” and the next preceding entity (“with” at `950:954`).

```
MENTION 1:
   entity        expr     pos
 DrugName  prednisone 269:279
 Strength         5mg 280:283
  DoseAmt           2 284:285
    Route    by mouth 294:302
Frequency       daily 303:308


MENTION 2:
      entity         expr     pos
DoseSchedule        Taper 818:823
    DrugName   prednisone 824:834
DoseStrength         10mg 836:840
   Frequency        daily 841:846
 Preposition          for 847:850
    Duration       7 days 851:857
  Transition         then 858:862
DoseStrength          5mg 863:866
   Frequency        daily 867:872
 Preposition          for 873:876
    Duration       7 days 877:883
  Transition         then 884:888
  DoseChange     decrease 889:897
 Preposition           by 898:900
DoseStrength          1mg 901:904
    Duration every 7 days 905:917
 Preposition        until 918:923
DoseSchedule         done 924:928
 Preposition           on 941:943
 Preposition         with 950:954
```

The preposition and transition entities also have some additional rules applied, considering how common the expressions in these dictionaries are (e.g., “of”, “to”, “then”, “;” or “,”). For both of these entities, the extraction must occur immediately adjacent to another extracted entity. Furthermore, for transition we require some sort of dose-relevant information to be extracted nearby (within 10 characters) on both sides. "Dose-relevant" entities include strength, dose amount, dose strength, frequency, intake time, duration, and dose schedule (for stopping keywords). Though less common, the same issue exists for the time keyword entity (common expressions include "was", "will", etc.). Rather than enforcing a character distance around time keyword extractions, we allow one word (of any length) to occur between time keyword and an adjacent entity.

Example: There are no further changes to mention #1. For mention #2, the prepositions "on" (`941:943`) and "with" (`950:954`) meet two separate conditions for removal; first, they are not immediately adjacent to another extracted entity and second, they occur at the very end of the group and we don't allow the last entity to be a preposition or transition. `medExtractR_tapering` combines any output from each drug mention, so the final output is:

```
      entity         expr     pos
    DrugName   prednisone 269:279
    Strength          5mg 280:283
     DoseAmt            2 284:285
       Route     by mouth 294:302
   Frequency        daily 303:308
DoseSchedule        Taper 818:823
    DrugName   prednisone 824:834
DoseStrength         10mg 836:840
   Frequency        daily 841:846
 Preposition          for 847:850
    Duration       7 days 851:857
  Transition         then 858:862
DoseStrength          5mg 863:866
   Frequency        daily 867:872
 Preposition          for 873:876
    Duration       7 days 877:883
  Transition         then 884:888
  DoseChange     decrease 889:897
 Preposition           by 898:900
DoseStrength          1mg 901:904
    Duration every 7 days 905:917
 Preposition        until 918:923
DoseSchedule         done 924:928
```

References

1. Weeks HL, Beck C, McNeer E, Williams ML, Bejan CA, Denny JC, Choi L. medExtractR: A targeted, customizable approach to medication extraction from electronic health records. Journal of the American Medical Informatics Association. 2020 Mar;27(3):407-18. doi: 10.1093/jamia/ocz207.

2. Choi L, Beck C, Weeks HL, and McNeer E (2020). EHR: Electronic Health Record (EHR) Data Processing and Analysis Tool. R package version 0.3-1. https://CRAN.R-project.org/package=EHR

3. Nelson SJ, Zeng K, Kilbourne J, Powell T, Moore R. Normalized names for clinical drugs: RxNorm at 6 years. J Am Med Inform Assoc. 2011 Jul-Aug;18(4)441-8. doi: 10.1136/amiajnl-2011-000116. Epub 2011 Apr 21. PubMed PMID: 21515544; PubMed Central PMCID: PMC3128404.