PhUSE EU Connect 2018

Paper PP03

Journey to the centre of the protocol: Explorations in text mining for study eCRF development

Rod Bower, Roche, UK

ABSTRACT

CRF development for clinical trial database build should be driven by the study protocol. Currently, a manual review of the protocol takes place to identify the data collection needs. Tasks include reviewing study assessments and procedures to translate them into the standard library equivalents.

A well-defined set of data standards, protocol templates and review guidelines greatly expedites the process. However, with today's ever more complex trial designs, this may be time consuming and lack consistency.

This paper explores the use of text mining techniques applied to a protocol, in regards to clinical data collection. The aim being to highlight which standard data collection forms are applicable and where possible, what datapoints are to be collected. R was used to read, process and visualize the protocol text, enabling relevant information retrieval. I highlight the approach taken with the most useful methods encountered. Challenges are outlined with proposed solutions.

INTRODUCTION

Text mining, also known as text analysis, is the process of deriving relevant information from text. The text mined is usually large and unstructured with the goal being to extract useful, structured content from it.

The method involves pre-processing the input text to give it some basic structure, addition of some derived linguistic features and the removal of others. Patterns are derived within the data followed by evaluation and interpretation of the output.

Typical text mining applications include text categorization (example spam or non spam emails), text clustering (document organization), concept/entity extraction (such as identifying people, places and organizations from documents), sentiment analysis (to identify and extract subjective information in documents), document summarization (automatically provide the most important points in the original document), and entity relation modelling (i.e. learning relations between named entities). In the pharmaceutical industry, text mining has multiple applications from drug discovery and clinical trial development to drug safety monitoring. Researchers can quickly analyze huge amounts of literature and clinical data to help inform and guide R&D.

Here I aim to leverage text mining techniques to pull information relevant to data collection from study protocols.

PRE-PROCESSING

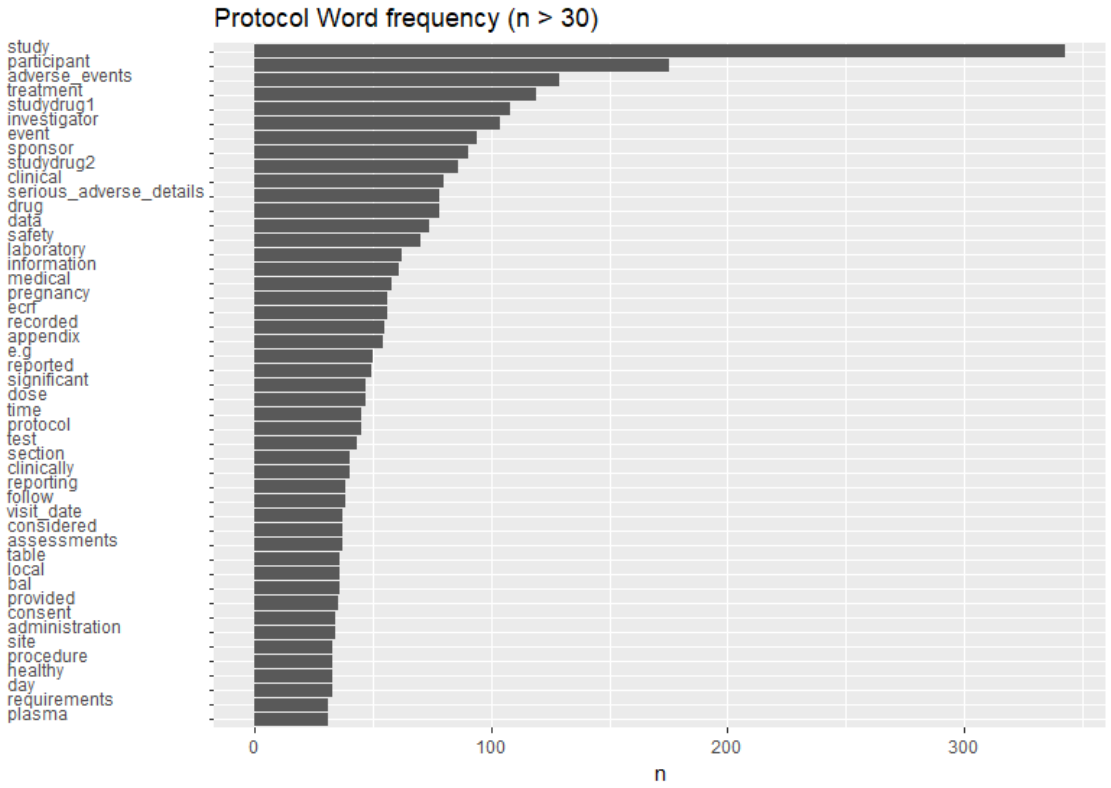
Although the study protocol is a detailed document that describes how a clinical trial will be conducted, it can be considered to be unstructured in that for Data Management, information related to CRF development/database build can be buried amongst data relevant to other stakeholders. Careful review by an experienced Data Manager is required to identify assessments, visits, lab tests etc. Sub sectioning of the document helps and tables showing the schedule of assessments are particularly useful but it's still a sizable body of text to sift through.

For these explorations, R (particularly the tidytext and tokenizers packages) was used to reshape the protocol text. The protocol was saved to a text file then read line by line into a single column dataframe, each line of the protocol in a row. The preprocessing continued by splitting each row into individual words. The result is a dataframe with a column of single words. Organizing the text in this format allows it to be manipulated, processed, and visualized for exploration. In certain cases, it makes sense to combine multiple words to preserve their meaning (for example 'adverse' followed by 'event' is more useful retained as adverse_event than split).

There are common words, known as 'stop words' that are not useful for analysis (e.g. the/of/to). Stop words can be removed by referencing a dataset of such words contained in the tidytext package. Further unwanted words can be removed the same way using a list of custom stop words. Finally, punctuation is removed and text converted to lowercase.

START TO EXPLORE

With the text now in a workable format we can start on some basic analysis. Showing the most common words in the text can reveal the documents content (i.e. the more times a word occurs, it can be suggested that the more significant it is to the document).



term frequency

Looking at the most popular individual words produces an interesting output. To gain further insight into the protocol text we may like to look into the relationships between words, examining which words tend to follow others. The n-grams setting looks at consecutive sequences of words, providing a measure of how often word X is followed by word Y. Looking at 2 consecutive words is known as bigrams.

	bigram	n
	<chr>	<int>
1	study treatment	60
2	study drug	46
3	clinically significant	32
4	adverse_events ecrf	31
5	bal procedure	24
6	drug administration	23
7	24 hours	19
8	informed consent	19
9	appendix 3	18
10	day 1	16

bigram count

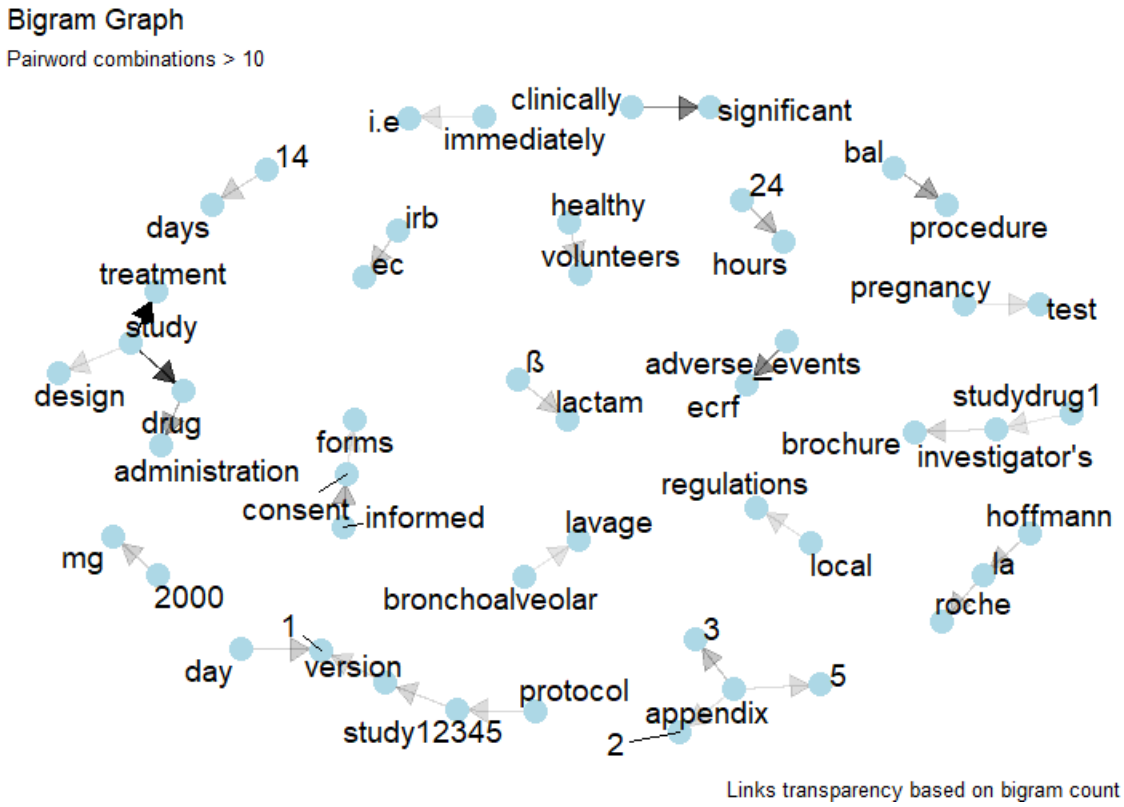
This can be extended to show the most common trigrams (three word consecutive sequences).

	trigram	n
	<chr>	<int>
1	study drug administration	21
2	hoffmann la roche	16
3	protocol study12345 version	13
4	study12345 version 1	13
5	studydrug1 investigator's brochure	11
6	epithelial lining fluid	9
7	sponsor immediately i.e	8
8	studydrug2 uspi 2016	7
9	2 protocol study12345	6
10	clinical laboratory tests	5

trigram count

VIZUALIZE WORDS AS A NETWORK

As an overview of the entire protocol, we may be interested in visualizing all of the relationships among words, not just the top few at a time. This can be achieved by arranging the words into a network, or as a graph (a combination of connected nodes). We show words connected, i.e. bigrams (word1 followed by word2 with the direction indicated with an arrow), the weight of the connection determined by the bigram count (higher counts shown by less transparent links). The igraph package is used for this with the result visualized with ggplot.



bigram graph

Note that to make the visualization presentable, only the most common connections are shown. By dialing down the frequency restriction, more of the complete text can be viewed.

COUNTING AND CORRELATING PAIRS OF WORDS

Exploring pairs of adjacent words provides a useful insight into the protocol. However, we may also be interested in words that occur within particular sections, even if they don't occur right next to each other.

To count and correlate among sections, the protocol text is divided into 10 line sections. The pairwise_count() function from the widyr package is employed to count common pairs of words appearing within the same section.

```
   item1         item2            n
   <chr>         <chr>        <dbl>
 1 participant   study          55.
 2 study         participant    55.
 3 study         treatment      51.
 4 treatment     study          51.
 5 study         investigator   42.
 6 study         drug           42.
 7 investigator  study          42.
 8 drug          study          42.
 9 study         clinical       38.
10 clinical      study          38.
# ... with 480,346 more rows
```

common word pairs

In this case the most common pair of words in a section are "study" and "treatment". To further investigate, we can determine what words most often occur with "Study" by simply filtering.

```
> word_pairs %>%
+   filter(item1 == "study")
# A tibble: 2,032 x 3
   item1 item2            n
   <chr> <chr>        <dbl>
 1 study participant    55.
 2 study treatment      51.
 3 study investigator   42.
 4 study drug           42.
 5 study clinical       38.
 6 study 2              35.
 7 study 1              34.
 8 study sponsor        30.
 9 study safety         30.
10 study time           29.
# ... with 2,022 more rows
```

word pairs occurring with the word 'study'

EXAMINING PAIRWISE CORRELATION

Pairs like “study” and “treatment” are the most common co-occurring words, but that may not be especially meaningful since they’re also in the top5 most common individual words.

Instead, we may want to take the approach of examining correlation among words, to indicate how often they appear together relative to how often they appear separately. The phi coefficient (similar to the Pearson correlation coefficient) is a measure of the degree of association between two binary variables. Here we measure how much more likely it is that either both word X and Y appear in a section, or neither do, than that one appears without the other. The pairwise_cor() function in widyr finds the phi coefficient between words based on how often they appear in the same section.

	item1	item2	correlation
	<chr>	<chr>	<dbl>
1	significant	clinically	0.920
2	clinically	significant	0.920
3	consent	informed	0.900
4	informed	consent	0.900
5	volunteers	healthy	0.870
6	healthy	volunteers	0.870
7	medications	previous_and_concomitant_medications	0.849
8	previous_and_concomitant_medications	medications	0.849
9	procedure	bal	0.842
10	bal	procedure	0.842

... with 10,090 more rows

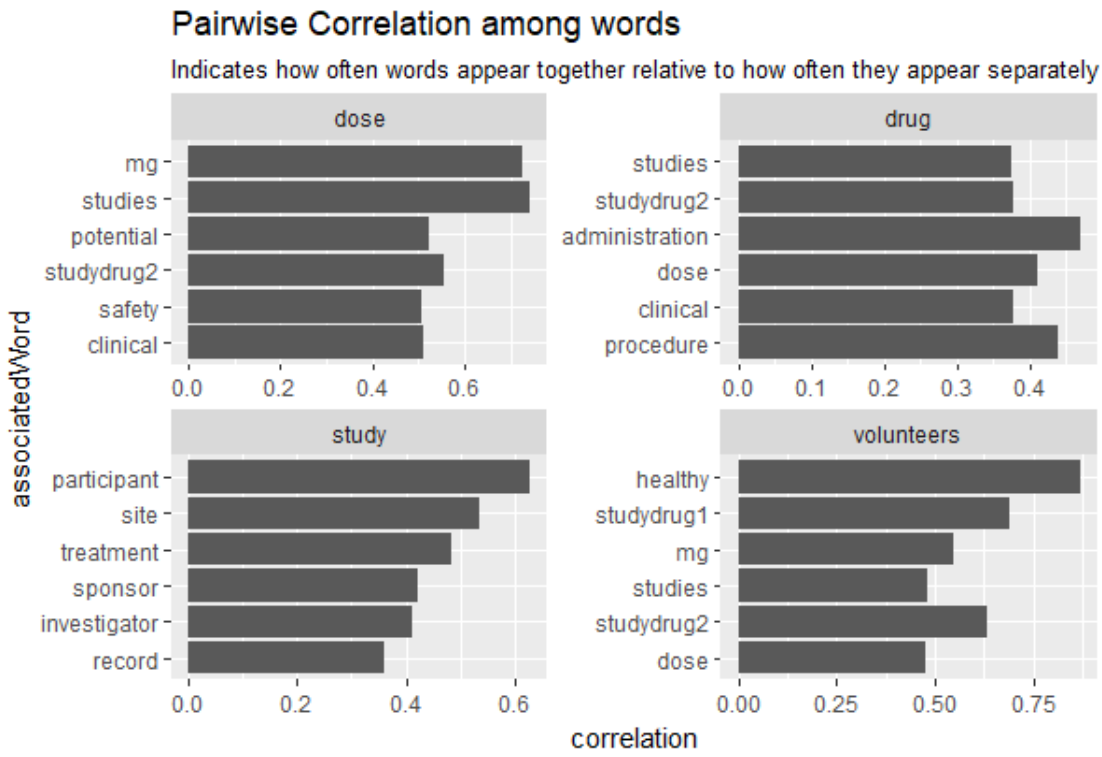
most correlated pairs

The output aids exploration. We could find the words most correlated with a word like “infusion” using a filter.

```
> word_cors_infusion %>%
+   filter(item1 == "infusion")
# A tibble: 100 x 3
   item1    item2          correlation
   <chr>    <chr>                <dbl>
 1 infusion single               0.551
 2 infusion related              0.361
 3 infusion ecrf                 0.327
 4 infusion time                 0.305
 5 infusion event                0.289
 6 infusion adverse_events       0.239
 7 infusion drug                 0.228
 8 infusion study                0.221
 9 infusion administered         0.215
10 infusion e.g                  0.212
# ... with 90 more rows
```

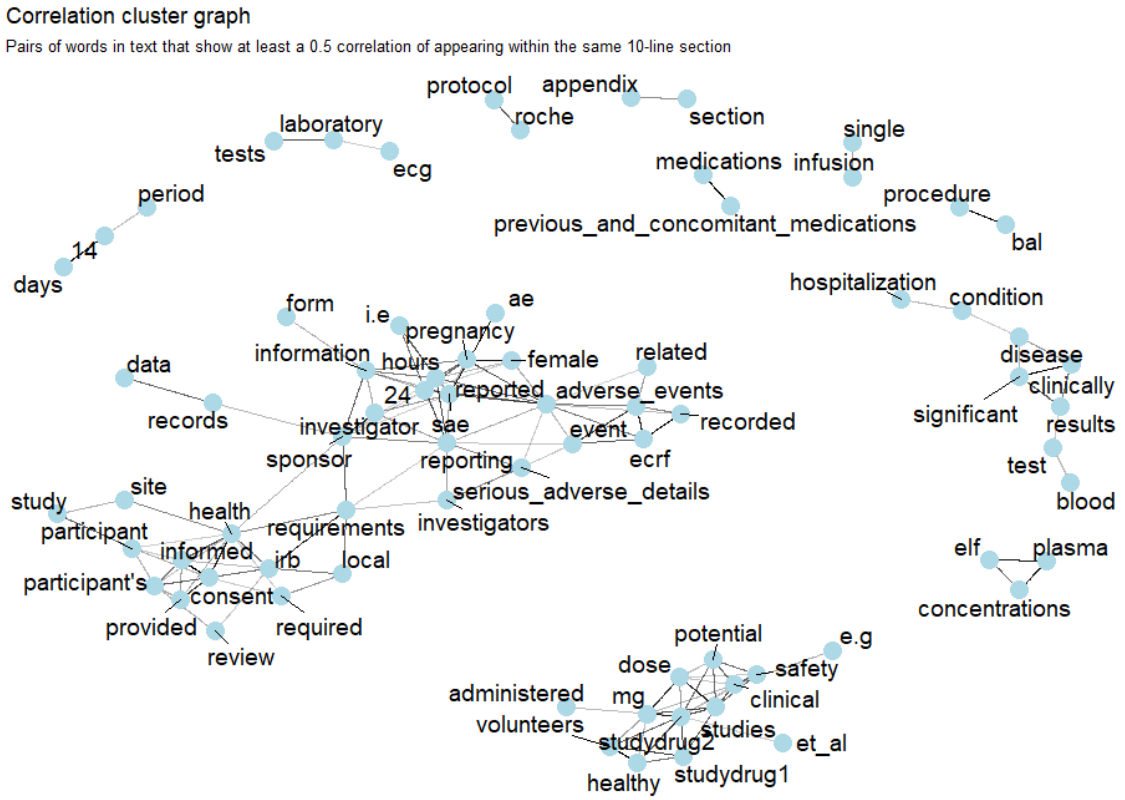
word correlation to ’infusion

We can pick particular ‘of interest’ words and find other words most associated with them.



specific word correlation

As with bigrams, we can visualize the correlations and clusters of words.



correlation graph

Note that unlike the bigram analysis, the relationships here are symmetrical, rather than directional (there are no arrows).

SUMMARY SO FAR

So far so interesting but what's useful. We've applied some techniques that highlight common words/pairs of words in the document. Without reading the protocol, elements of the study have risen to the surface. For example; that the example study involves healthy volunteers, a pregnancy test and the study medication is given as an infusion. We also see terms that are no surprise. For example, we would expect to see consent and adverse events mentioned in a protocol. The fact that there is information we are expecting provides an angle to produce something useful regarding CRF build (the paper's stated aim).

If we can produce a comprehensive bag of expected possible words that relates to data collection, could we target those. Perhaps just finding a single occurrence of them in the protocol text is a strong indication they should be in the study build. In fact, if you have a set of data collection standards, the target words have already been identified for use in this manner.

IDENTIFY STANDARD FORMS

We would expect to find all the assessments to be performed in the study, detailed in the protocol. This would be as part of the table/schedule of assessments or in detail in the protocol text. So searching for standard library form names in the text should identify the those required for the study build. The study data manager/database programmer, would identify these manually so the process is familiar.

After pre-processing, individual terms are retained from the protocol and using a suitable join are matched with the library list of terms. The initial results show matches but fewer than we'd expect. Some assessments simply are not listed in the protocol text verbatim. Perhaps some further processing of the library target list will up the number of matches.

An approach to this additional preprocessing is to compile a dictionary of library assessments and use to make the protocol text and library forms uniform. This is the same as the interpretation step that goes on in the head of the manual reviewer.

This includes:

- Combining relevant multiple word terms (e.g. the words 'Subject' and 'characteristics' remain together when consecutive).
- Replacing alternative names or shortenings (e.g. ECG/Electrocardiogram).
- Use of indirect matching to indicate an assessment where the actual term may not be listed in the protocol. An example of this is that height/weight may be referred to in the text rather than the assessment term 'anthropometric measurements'.

A join of the protocol and library terms is used to highlight the standard assessments required by the study.

1	adverse_events	129
2	pregnancy	56
3	visit_date	37
4	subject_disposition	10
5	vital_signs	10
6	medical_history	8
7	electrocardiogram	6
8	enrollment	6
9	hematology	4
10	blood_chemistry	3
11	urinalysis	3
12	anthropometric_measurements	2
13	risk_factors	2
14	serology	2
15	surgery_and_procedure_history	2
16	demography	1
17	subject_characteristics	1

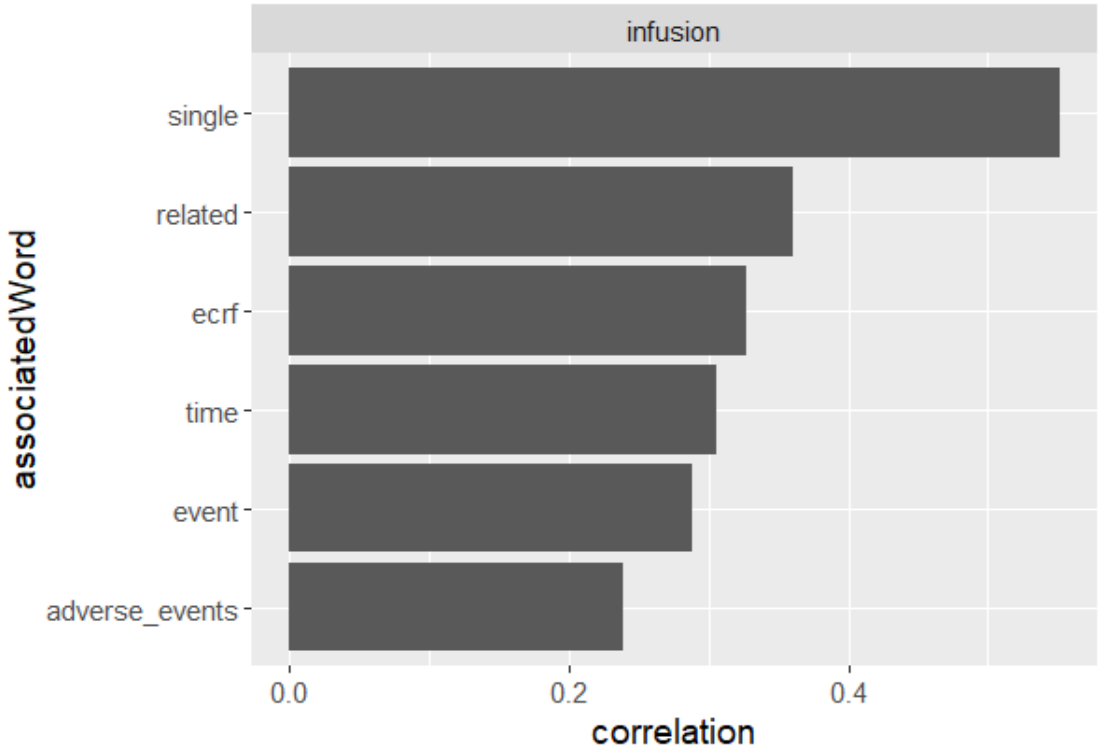
assessments

Note that some operational only forms may be needed for data collection but it is unlikely these will be listed in the protocol. This may be overcome by indirect matching/including as required.

IDENTIFY TYPE OF FORM REQUIRED

For certain assessments, the library contains different versions. For example, the study drug administration form has two versions. One for oral admin another for infusions or injections. Use of the paring/correlation technique discussed earlier can provide a firm clue on which applies. After correlating the text, the form version is detected by filtering on the 3 expected options (oral, infusion and injection) as the words of interest.

Only the infusion filter produced a result in this case, so the 'infusion/injection' version of the CRF form should be used for this study.



drug admin form filter

FIELDS WITHIN A FORM

Identifying individual datapoints to be captured within an assessment adds another level of complexity. This can be attempted using term matching within form standards. This example targets laboratory assessments aiming to select the required chemistry parameters. The lab parameters are read in from the chemistry form to refer to. A chemistry specific dictionary is also referenced. This is used to convert to a preferred text. Useful in combining multiple word parameters (e.g. creatine kinase to creatine_kinase) and alternate common naming/acronym's (AST = aspartate aminotransferase). The concept is the same as that detailed for forms. The dictionary is run over the protocol text and library text to create a common use terminology. A fuzzy join of the protocol text to the chemistry library gives us a list of the analytes found.

	word	n
	<chr>	<int>
1	urea	10
2	alanine_aminotransferase	8
3	aspartate_aminotransferase	8
4	repeat	6
5	potassium	5
6	alkaline_phosphatase	3
7	sodium	3
8	total_bilirubin	3
9	chloride	2
10	glucose	2
11	albumin	1
12	bicarbonate	1
13	calcium	1
14	creatinine	1
15	direct_bilirubin	1
16	lactate_dehydrogenase	1
17	phosphate	1

blood chemistry parameters

In addition to labs, parameters can be identified from other well defined assessments such as ECG's.

CONCLUSION

These explorations demonstrate that the use of text mining techniques can be useful in retrieving data collection information from the protocol to assist protocol review. Much refinement would be needed for it to be relied upon without additional manual efforts. The techniques employed relied heavily on an established set of standards, targeting what may be collected and developing the pre-processing dictionary (the dictionary being the key to interpreting the text). Extending the scope beyond this to identify non standards would be difficult using the approach tried here. Looking forward, a clinical trial protocol template containing a common structure and language connected to CDISC data standards (such as TransCelerate's Common Protocol Template Initiative) would ease and increase the effectiveness of this task.

REFERENCES/FURTHER READING

Silge J and Robinson D (2016). "tidytext: Text Mining and Analysis Using Tidy Data Principles in R." *JOSS*, *1*(3). doi: 10.21105/joss.00037 (URL: http://doi.org/10.21105/joss.00037), <URL: http://dx.doi.org/10.21105/joss.00037>.

Lincoln Mullen (2018). tokenizers: Fast, Consistent Tokenization of Natural Language Text. R package version 0.2.0. https://CRAN.R-project.org/package=tokenizers

Csardi G, Nepusz T: The igraph software package for complex network research, InterJournal, Complex Systems 1695. 2006. http://igraph.org

H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2009.

David Robinson (2018). widyr: Widen, Process, then Re-Tidy Data. R package version 0.1.1. https://CRAN.R-project.org/package=widyr

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Rod Bower
Roche Products Ltd
6 Falcon Way, Shire Park
Welwyn Garden City / AL7 1TW
rodrick.bower@roche.com