Detecting vaccines & therapeutics for COVID-19 with dictionaries & rules

Annie Tallund
Lund University
Faculty of Engineering
Sweden
an0284ta-s@student.lu.se

Sofi Flink
Lund University
Faculty of Engineering
Sweden
bmp13sfl@student.lu.se

Abstract

Natural Language Processing (NLP) has been proven to be an efficient way for researchers to summarize the scientific knowledge in various fields. Due to the sudden emerged crisis of the COVID-19 pandemic, the need is greater than ever for researchers to be able to summarize the knowledge available within the scientific community. This is a sub-project to generate an outline of what the scientific field knows about therapeutics and vaccines for viruses, particularly COVID-19, and in a bigger scope the goal is to summarize all knowledge about COVID-19 in order to discover solutions and strategies to combat the pandemic. The tool developed uses dictionaries and rules in order to filter out articles related to COVID-19. This tool should hopefully be helpful aid for researchers to understand the field of COVID-19 though future iterations is necessary.

1 Introduction

The last few years with the exploding information age the research community in all fields have been flooded with new publications of articles and documents. It is no longer possible for researchers to keep track of all relevant publications and as such understand their field in its entirety, thus no complete picture of the scientific knowledge exists. This difficulty has been even more apparent as a result of the sudden growing crisis of the COVID-19 pandemic. Only counting the past few months more than 52,000 articles, and probably more at the time of writing, have been published. This article describes a tool in Natural Language Processing (NLP) developed for summarizing and giving researchers a way to gain a complete picture of the scientific knowledge regarding COVID-19 and its vaccines and therapeutics.

This project was developed together with 4 other groups, where the focus of this paper and our group contribution was using dictionaries and rules for generating Pubannotations for researchers to use in their research. By defining a set of words that should be recognized, the tool can be used to find articles that might be relevant for the person using it. The dictionary-rules tagger was also evaluated using our own implementation of an evaluator class generating precision and recall values, micro and macro figures and harmonic means.

2 Previous work

Using a dictionary-based approach in NER can be a complex task. Since a word can have different meanings depending on context, the technique can result in false positives. In addition, tokens and different spelling might have the model miss to identify some words that should have been recognized (false negatives). The issue has earlier been dealt with by combining a number of strategies, which improved the performance on biomedical entities (Song et al., 2015). Using a minimum edit distance approach means that two strings are compared on how different they are, namely how many characters have to be edited to obtain the optimal match. This can account for words that are spelled differently, by comparing the word in the dictionary with the token from the text. The article also described how part-of-speech (POS) tagging was utilized to obtain syntactical properties of words, which can be meaningful when the dictionary contains words that have different meanings and hence account for the false positives that are generated.

A more advanced approach was made by researchers in Korea, as they developed the tool bioBERT, which is a pre-trained language representation model for biomedical text mining (Lee et al., 2019). It has been trained on a large corpora of biomedical texts, and works with the Bidirectional Encoder Representations from Transformers

(BERT) technique developed by Google for NLP. So instead of using dictionaries, it takes a more general approach where a lot of biomedical phrases have been included in the pre-training. This model can work with relationship extraction, which is also an important part of NER, and specifically for biomedical text mining. Therefore, the bioBERT contributes with an interesting perspective. However, rather than a different approach, is is more of a future step in the process of NER, and thus can be combined with our work to develop better tools for detecting vaccines and theurapeutics for COVID-19.

3 Implementation

We have produced several different iterations of our tagger during the course of this project. The first iteration was a simple script for the tagging process and generation of PubAnnotation-files. This was sufficient to produce results, but since someone else will be picking up where we left off, we focused on modular and easy to read code. By splitting up the code in different classes depending on their purpose and utility it will be easier for future programmers to continue developing new iterations of the tagger.

3.1 Program application overview

The program as a whole uses input data consisting of articles in JSON format, a metadata file in CSV format containing relevant information of the articles for generating PubAnnotations, and dictionaries for different word classes containing words to be tagged in the articles. The program processes each text section one by one. By using regular expressions of the words in the dictionaries and rule patterns (in current iteration all words ending with 'vir' as 'chemical_antiviral) it will find matches in the text and retrieve their spans for the corresponding text section. These matches are collected and used to construct denotations in order to generate proper PubAnnotations for each text section in each article. After generating the files and exporting them to an output folder the evaluator will run and review the output. Since there is no gold standard PubAnnotation result for all articles the evaluation was only done on selected few articles where gold standard PubAnnotations had been created.

3.2 File format and executing the code

The input files of the articles are JSON-files from which the program using 'json' library will process to dictionaries.

```
{
   "paper_id": "31996494",
   "metadata": {
      "title": "Drug treatment options
          for the 2019-new coronavirus
          (2019-nCoV). ",
      "authors": []
   },
   "abstract": [
      {
         "text": "As of January 22
             (...)",
         "cite_spans": [],
         "ref_spans": [],
         "section": "Abstract"
      }
   ],
   "body_text": []
}
```

Listing 1: Input file example

The output files are generated with 'PubannotationGenerator" which uses input argument a dictionary of PubAnnotations and directory path for the output files. The annotation strings are generated and quotations in texts escaped, every text section will generate it's own file and be exported to designated directory.

```
{
   "cord_uid":"31996494",
   "sourcedb":"PMC",
   "sourceid":"PMC31996494",
   "divid":1,
   "text":"As of January 22 (...)",
   "project":"cdlai_CORD-19",
   "denotations": [
      {
         "id":"Virus_SARS-CoV-2",
         "span":{"begin":57,"end":72},
         "obj":"AS-dictionary_T3"
      },
      {
         "id":"chemical_antiviral",
         "span":{"begin":825,"end":835},
         "obj":"AS-dictionary_T3"
      },
      {
         "id":"chemical_antiviral",
         "span":{"begin":1134,"end":1145},
         "obj":"AS-dictionary_T3"
      }
   ]
}
```

Listing 2: Output file example

Lastly, the results of the tagger are evaluated against a gold standard. It obtains the precision and recall for the tagger when comparing the gold standard PubAnnotations and tagger output.

The program can be run run though the terminal from the bin project directory with the following command:

```
python3 -m bin.main
```

It can also run with an IDE where the working directory should be assigned to 'edan70' and the source folder to 'edan70/src'. The python library 'pandas' will have to be installed on the system before running the program.

3.3 DictionaryTagger

The tagger uses dictionaries and rules to tag articles. In current iteration, rule word class 'chemical_antiviral' and dictionary word classes 'Disease_COVID-19', 'Symptom_Covid-19' and 'Virus_SARS-CoV-2' are supported. It is possible to add more dictionaries as input to the tagger, but there are no specialized rule priority implemented for any other than the three dictionaries mentioned above at the current state of the tagger.

3.3.1 Data structures

The program uses a number of data structures to keep track of the data. The dictionaries are interpreted as a list of words, which are saved as a dictionary with their file name as key (being their word class id) and the list of word as the value pair. For the rules they have to be hard coded in to the tagger, since there is no automated way to translate human instructions to a regular expression. The tagger supports matching of all words ending in 'vir', and for this the following python regular expression was used:

```
r′(?i)§⋆vir′_
```

The metadata is a CSV file, which with the 'pandas' library processes each row to a dictionary with the header as key for the column values. The metadata dictionaries are saved in to a list. In order to get O(1) time complexity when finding the correct metadata corresponding to a certain article, another dictionary is used to map list indices to corresponding metadata article dictionary as value together with their paper id/sha. This also solves the edge case for when an article in the metadata has two unique ids, while the JSON article files always have one.

The JSON article files are processed with the python 'json' library to dictionaries and are saved in a python dictionary with their unique 'cord_uid' as key, The dictionaries with words are represented as 'vocabularies' in the syntax so to avoid confusion with the python data structure dictionaries.

3.3.2 Tagging algorithm

The program iterates through article dictionaries and retrieves the texts. Each paragraph is taken and is sent into the `tag_paragraph` method. The method iterates through each vocabulary with words and constructs a regular expression for each. The regular expression is used to search for all occurrences in the paragraph, and returns match objects with span of the match in the text sequence. The matches are collected and compared, with double annotation matches following the following rules:

- For double annotations of the same class the match with more characters is retained.
- If a string is found in virus_sarscov2 and disease_covid19 the match with more characters is retained, e.g. SARS-Cov2 infection is tagged as disease_covid19 even though it also contains the virus name.

The regular expression has to look for matches case-insensitive, plural versions of word, combinations of hyphens between non compound words, some matches use white space following a hyphen. Following is the regular expressions appended to each word in the vocabularies:

```
case_insensitive_reg = r'(?i)'

hyphen_or_wp_or_both_reg
   = r'(\s|\-\s?)'

opt_plural_reg = r'(es|s)?'

boundary_reg = r'\b'

composite_word += composite_word
   + hyphen_or_wp_or_both_reg

word_regex = += composite_word
   + opt_plural_reg
   + boundary_reg
```

Listing 3: Regular expressions

3.4 PubannotationGenerator

The class takes dictionaries of the articles

```
{file_name : pubannotation_dict}
```

which is obtained from `DictionaryTagger`, and constructs strings with denotations for each key in the dictionary. Then, all denotations are concatenated and stored in a annotation string. Lastly, the string is exported into its own file.

3.5 PubannotationEvaluator

The evaluator takes directory paths to tagger output files and gold standard Pubannotations files as argument. It compares their denotations: the span and the word class (dictionary). Once the four categories *true positives*, *false positives*, *true negatives* and *false negatives* are calculated, the precision and recall is calculated for each class. Also, the class does a *macro* and *micro* interpretation of the results and harmonic mean value, which is all put together and presented in the prompt.

4 Method

For the purpose of the project, article data sets where needed. These where retrieved from KAGGLE COVID-19 project, CORD-19, consisting of 52,000 articles. For the purpose of easier testing, a selection of a 100 articles where retrieved and a script was implemented for receiving correct columns of medata from the metadata csv file. In order to evaluate the tagger a gold standard was needed which was given by Aits Labs consisting of 10 articles.

4.1 Tagging

The tool uses regex expressions to find matches of words in the dictionaries. In order to find all the matches correctly, it takes a few special cases into account.

For one part there are similar phrases in the dictionaries, which can cause obsolete tagging, for example "*new coronavirus acute illness*" and "*2019 new coronavirus acute illness*". To solve this, we save only the match that contains the most characters by saving all matches and comparing them. Secondly, there could be one phrase that occurs more than once in a text. This can be missed if the taggers registers a match connected to the word itself and not the span where it was obtained. Therefore, we use the span and the dictionary to represent a match.

Another category of potential errors is the actual formatting of the texts where we conduct the search. For example, a phrase can be missed if it includes hyphens that is not included in the dictionary, for instance "*2019-new coronavirus*" and "*2019 new corona virus*". This was solved by creating a regular expression appended between each word in a non compound word for optional whitespace or hyphen or a combination of both. Also, the search needs to be case insensitive in order to obtain matches with different formats such as "*Covid-19*" and "*COVID-19*". As mentioned earlier, plural versions of all words are also tested. That some words are represented nonsensical does not matter for the tagging process since the paragraphs contain properly written article texts.

4.2 Evaluation

The evaluator compares the tagger's output files to a gold standard or *true* output. By going through all the spans in the denotations, and registering any match, we can obtain figures for precision P and recall R. These are calculated from the number of true positives tp, false positives fp and false negatives fn. This gives us a perception of the relevance of the tags we obtain with the taggers, and how well it correlates with the true model (Sammut and Webb, 2010).

$$P = \frac{tp}{tp + fp} \tag{1}$$

$$R = \frac{tp}{tp + fn} \tag{2}$$

Precision measures how many of the retrieved matches are relevant, while recall tells us how well we retrieved relevant matches. For example, a tagger can have a lot of matches but not the ones that we wish to have (based on the values that we know are in the *true* set). This scenario will give a low precision. On the other hand, we can can have a lot of correct matches, but having missed a lot of them that are in *true* set, which would give a low recall. If the words we tag are relevant, and we don't miss any matches that are supposed to be there, the *accuracy* is high.

We also calculate the harmonic mean, the F_1 score. This is a way of averaging the precision and recall, and gives a perception of the overall accuracy of the tagger (Sammut and Webb, 2010).

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3}$$

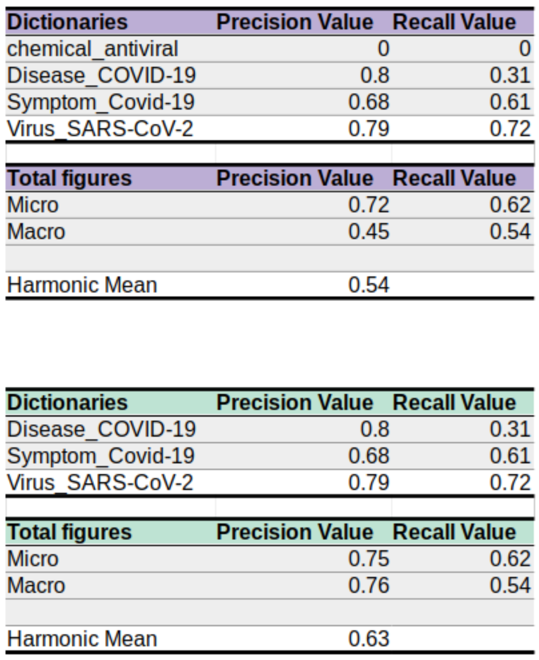
In addition, we do a micro and macro evaluation of the performance. The macro computes the metrics for each class separately and take the average. The micro will collect the results for each class and aggregate the contributions (Olivas et al., 2009).

5 Results

The table below 1 is a presentation of all the collected match values for each word class. The word class 'chemical_antiviral' is excluded, since it dilutes the score as lots of false positives, due to the gold standard used did not support the word class at the time. We can see that the different classes have similar precision, and that the recall values differ slightly. The micro and macro values are close to identical. The over-all harmonic mean ended up being 63%, after adjusting for the class that was not contained in the golden standard.

These are the row figures in the gold standard for each class respectively

Class	Total	TP	FP	FN
Disease_COVID-19	13	4	1	9
Symptom_COVID-19	28	17	8	11
Virus_SARS-CoV-2	46	33	9	13
Total	87	54	33	18



Dictionaries	Precision Value	Recall Value
chemical_antiviral	0	0
Disease_COVID-19	0.8	0.31
Symptom_Covid-19	0.68	0.61
Virus_SARS-CoV-2	0.79	0.72

Total figures	Precision Value	Recall Value
Micro	0.72	0.62
Macro	0.45	0.54
Harmonic Mean	0.54	

Dictionaries	Precision Value	Recall Value
Disease_COVID-19	0.8	0.31
Symptom_Covid-19	0.68	0.61
Virus_SARS-CoV-2	0.79	0.72

Total figures	Precision Value	Recall Value
Micro	0.75	0.62
Macro	0.76	0.54
Harmonic Mean	0.63	

Figure 1: Precision & recall result with harmonic mean.

The time was measured for running the programs using the 100 subset of the articles. The following was the result retrieved:

- Run-time is 4 hours 41 minutes 52 seconds
- Running the program using 'Spyder' environment, the tagging process, as expected, consumed the most time. In section 'Future Iterations' optimization suggestions are made to the algorithm.

6 Discussion

One interesting aspect of the results is that the macro precision value is lower than the micro equivalent. An explanation of this can be the tagged classes. The *chemical_antiviral* is not included in the gold standard, which means the performance of this class will be zero, leading to a so called class imbalance. Since the macro averages all the classes, the result of the missing dictionary will lower that average. In micro, however, the results will be aggregated and averaged differently, and thus the result will be slightly more balanced. This is also apparent in the second figure where the micro and macro precision is similar. This also tells us that the classes are balanced, meaning the performance and occurrence of matches is similar for most of the classes.

This issue is also the reason we decided to make two evaluation tables: one with the *chemical_antiviral* included, and one without. This dictionary is not per definition a dictionary, but rather a pattern that we are looking for in the corpus. In the second iteration of the testing, the figures are increasing, which was expected. The lesson learned is to base the evaluation on data with the same constraints and classes as the test set.

Despite all the special cases that are treated in the tagger, the evaluation shows us that there is room for improvement. When going through the matches manually we discovered some possible error sources. Almost all of the missed matches was a problem of having *incomplete dictionaries*. It means that there existed word variations for the corona virus that are not contained in the dictionaries, that the gold standard had identified. This problem is solved by manually completing the lists with new words that should be tagged. The result is also a reflection of the precision and recall values: the precision values are similar for the relevant classes, however not close to 100%, and recall differs. In other words, we have more words missing in the classes with lower recalls, as the gold standard highlighted more words than our tagger was able to detect.

In our tagger we had an interesting corner case: the word "*corona virus*" was intentionally not put into any of the dictionaries, because this is an umbrella term for different types of viruses that are not necessarily related to the COVID-2019. Choosing what words to have in the dictionary can therefore be a complex task of finding the exact variations that are relevant.

6.1 Difficulties

It is a lot of manual work in order to confirm the result, and even manual work is prone to error. When confirming the result for tagging the articles in the gold standard, we manually compared all the denotations and described all discrepancies in the file 'Supplemental file' included in the 'Appendix'. It helped resolve some edge cases we had missed in the process.

We constantly had to rework our code in order to create readable syntax and documentation for all our methods and classes. For future programmers developing the code all parts of the codes intention had to be clear to the reader. There is probably further work here to be done.

6.2 Future Iterations

The current implementation of the tagger runs on polynomial time, but the running time will grow with bigger data sets. One suggestion for future work on the project is therefore to optimize the running time. One suggestion is to pre-process and categorize the dictionaries further, and use of stop words, so that any unlikely matches are excluded. A more trivial idea is to complement the dictionaries with any words that are in the gold standard. Both require manual work. Similarly, adding more dictionaries would be another way of extending the possible uses to identify more words.

A further suggestion for optimizing the run time is including all words in aĺl dictionaries in one ordered list. Most of the prioritization rules depend on most characters match, so the ordered list will start with the longest words. The next element will contain a string of the words word class. As such the iteration of words will be done with "`for x in range(0, len(word_list) + 1, 2)`". The first match found will be the word to be tagged, and the word class of that world is easily retrieved. The current implementation iterates through all the articles for every dictionary and rule. This complexity can be reduced to one iteration with this algorithm.

Also, if the project as a whole was to be developed further, it would be interesting to have a look at words in their context and rate the relevance. In our case, this would solve the issue of not having the umbrella term "*corona virus*" in the dictionaries, as we could detect the context. Even if that is not a trivial task, it is one way to work around something that would add value to the tool, namely using relevant terms that can have different meanings.

Acknowledgements

This project has been adapted from the instructions of supervisors Sonja Aits, reserach manager of Cell Death and Lysosomes at Lund University Faculty of Medicine, and Pierre Nugues, professor at Department of Computer Science with focus area Computational Linguistics. This project uses the English dictionaries generated by the toolbox developed by Sonja Aits and her team, gold and silver standard corpora for bioNLP related to SARS-CoV-2 and COVID-19

References

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. *Biobert: a pre-trained biomedical language representation model for biomedical text mining.*

Emilio Soria Olivas, Jose David Martin Guerrero, Marcelino Martinez Sober, Jose Rafael Magdalena Benedito, and Antonio Jose Serrano Lopez. 2009. *Handbook Of Research On Machine Learning Applications and Trends: Algorithms, Methods and Techniques - 2 Volumes.* Information Science Reference - Imprint of: IGI Publishing, Hershey, PA.

Claude Sammut and Geoffrey I Webb, editors. 2010. *Encyclopedia of Machine Learning.* Springer US, Boston, MA.

Min Song, Hwanjo Yu, and Wook-Shin Han. 2015. *Developing a hybrid dictionary-based bio-entity recognition technique.* BMC Medical Informatics and Decision Making 15, S9.

7 Appendix

7.1 Supplemental file

```
31996494 Abstract:
    - Virus_SARS-CoV-2:
            Tagger result           Gold standard output        Comment
        *   begin:52    end:72      begin:52    end:72      '2019-new coronavirus'
        *   begin:74    end:83      begin:74    end:83      '2019-nCoV'
        *   begin:345   end:354     begin:345   end:354     '2019-nCoV'
        *   begin:1095  end:1104    begin:1095  end:1104    '2019-nCoV'
        *   begin:1158  end:1168    begin:1158  end:1168    '2019- nCoV'

    WARNING: NOTE THAT Disease_COVID-19 IS USED TO TAG '2019-nCoV' IN PMC7003341
ABSTRACT

    - Virus_family
        Tagger does not support this word class
    - chemical_antiviral
        Gold output does not support this word class

31996494 Title:
    - Virus_SARS-CoV-2:
            Tagger result           Gold standard output        Comment
        *   begin:31    end:51      begin:31    end:51      '2019-new coronavirus'
        *   begin:53    end:62      begin:53    end:62      '2019-nCoV'
32013309 Abstract:
    - Virus_SARS-CoV-2:
            Tagger result           Gold standard output        Comment
        *   N/A                 begin:0     end:11      Gold standard output tags 'Coronavirus'
which does
                                            not exist in dictionary
    - Disease_other
        Tagger does not support this word class
    - Virus_family
        Tagger does not support this word class
    - Symptom_COVID-19:
            Tagger result           Gold standard output        Comment
        *   begin:344   end:356     begin:344   end:356     'inflammation'
        *   begin:363   end:368     begin:363   end:368     'fever'
        *   begin:370   end:375     begin:370   end:375     'cough'
        *   N/A                 begin:415   end:445     Gold standard output tags 'dysfunction of
internal
                                            organs' which does not exist in dictionary
        *   N/A                 begin:463   end:468     Gold standard output tags 'death' which
does not
                                            exist in dictionary
        *   begin:1012  end:1024    begin:1012  end:1024    'inflammation'
    - Protein
        Tagger does not support this word class

32013309 Title:
    - Virus_family
        Tagger does not support this word class
```

- Symptom_COVID-19:

	Tagger result		Gold standard output		Comment
*	begin:45	end:57	begin:45	end:57	'inflammation'

PMC6988272 Abstract:

- Virus_SARS-CoV-2:

	Tagger result		Gold standard output		Comment
*	begin:2	end:19	begin:2	end:19	'novel coronavirus'
*	begin:21	end:30	begin:21	end:30	'2019-nCoV'

- Disease_COVID-19:

	Tagger result		Gold standard output		Comment
*	N/A		begin:40	end:72	Gold standard output tags 'severe acute respiratory disease' which does not exist in dictionary

PMC6988272 Title:

- Disease_COVID-19:

	Tagger result		Gold standard output		Comment
*	begin:73	end:101	begin:73	end:101	'novel coronavirus infections'

PMC7003341 Abstract:

- Disease_COVID-19:

	Tagger result		Gold standard output		Comment
*	N/A		begin:20	end:44	Gold standard output tags 'new type viral pneumonia disease' which does not exist in dictionary
*	N/A		begin:460	end:488	Gold standard output tags '2019-nCoV infected pneumonia' which does not exist in dictionary
*	N/A		begin:1062	end:1071	Gold standard output tags '2019-nCoV' which does not exist in dictionary. NOTE THAT Virus_SARS-CoV-2 IS USED TO TAG '2019-nCoV' IN 31996494 BUT NOT Disease_COVID-19
*	N/A		begin:1160	end:1188	Gold standard output tags '2019-nCoV infected pneumonia' which does not exist in dictionary.
*	begin:1239	end:1259	begin:1239	end:1259	'2019-nCoV infections'

- Virus_SARS-CoV-2:

	Tagger result		Gold standard output		Comment
*	begin:102	end:124	begin:102	end:124	'2019 novel coronavirus'
*	N/A		begin:126	end:135	Gold standard output tags '2019-nCoV' with paranthesis which tagger seems to miss
*	begin:1499	end:1508	begin:1499	end:1508	'2019-nCoV'

- Disease_other

  Tagger does not support this word class

- Symptom_COVID-19:

	Tagger result		Gold standard output		Comment

```
        *   begin:35     end:44      N/A                  Tagger tags 'pneumonia' present in
                                                  dictionary, gold standard output does not.
        *   begin:479    end:488     N/A                  Tagger tags 'pneumonia' present in
                                                  dictionary, gold standard output does not.
        *   begin:1179   end:1188    N/A                  Tagger tags 'pneumonia' present in
                                                  dictionary, gold standard output does not.

PMC7003341 Title:
    - Disease_COVID-19:
            Tagger result          Gold standard output       Comment
        *   N/A                 begin:60     end:113      Gold standard output tags '2019 novel
coronavirus                                                 (2019-nCoV) infected pneumonia'
which does not exist                                              in dictionary
    - Symptom_COVID-19:
            Tagger result          Gold standard output       Comment
        *   begin:104    end:113     N/A                  Tagger tags 'novel coronavirus' present in
                                                  dictionary, gold standard output does not.
    - Virus_SARS-CoV-2:
            Tagger result          Gold standard output       Comment

        *   begin:60     end:82      N/A                  Tagger tags '2019 novel coronavirus' as in
dictionary
                                                  but golden output has another longer match in
                                                  Disease_COVID-19 '2019 novel coronavirus
(2019-nCoV) infected pneumonia'.
        *   begin:84     end:93      N/A                  Tagger tags '2019-nCoV' as in dictionary
                                                  but golden output has another longer match in
                                                  Disease_COVID-19 '2019 novel coronavirus
(2019-nCoV) infected pneumonia'.

PMC7033720 Abstract:
    - Disease_other
        Tagger does not support this word class
    - Symptom_COVID-19:
            Tagger result          Gold standard output       Comment
        *   begin:43     end:52      N/A                  Tagger tags 'pneumonia' present in
dictionary, gold                                                standard output does not. Gold
standard output
                                                  tags it under Disease_othter instead with same span
        *   begin:356    end:361     begin:356    end:361
        *   begin:363    end:368     begin:363    end:368
    - Virus_SARS-CoV-2:
            Tagger result          Gold standard output       Comment
        *   begin:640    end:657     begin:640    end:657     'novel coronavirus'
        *   begin:665    end:674     begin:665    end:674     '2019-nCoV'
        *   begin:1068   end:1077    begin:1068   end:1077    '2019-nCoV'
        *   begin:1488   end:1497    begin:1488   end:1497    '2019-nCoV'
    - Virus_family
        Tagger does not support this word class
    - Virus_other
```

```
            Tagger does not support this word class
        - Protein
            Tagger does not support this word class

PMC7033720 Title:
        - Virus_SARS-CoV-2:
                Tagger result              Gold standard output          Comment
            *     N/A                      begin:37      end:60        Gold standard output tags 'novel human
coronavirus'
                                                                    which does not exist in dictionary'
        - Disease_other
            Tagger does not support this word class
        - Symptom_COVID-19:
                Tagger result              Gold standard output          Comment
            *     begin:81      end:90       N/A                       Tagger tags 'pneumonia' present in
dictionary, gold                                                                standard output does not.
Instead it taggs it for
                                                                    dictionary Virus_other
PMC7054940 Abstract:
        - Virus_family
            Tagger does not support this word class
        - Disease_other
            Tagger does not support this word class
        - Virus_family
            Tagger does not support this word class
        - Virus_other
            Tagger does not support this word class
        - Virus_SARS-CoV-2:
                Tagger result              Gold standard output          Comment
            *     N/A                      begin:375     end:398       Gold standard output tags 'This decade's
first CoV'
                                                                    which does not exist in dictionary
            *     begin:406     end:415      begin:406     end:415       '2019-nCoV'
            *     begin:902     end:911      begin:902     end:911       '2019-nCoV'
            *     N/A                      begin:978     end:989       Gold standard output tags 'novel virus'
which does
                                                                    not exist in dictionary
            *     N/A                      begin:1003    end:1007      Gold standard output tags 'nCoV' which
does
                                                                    not exist in dictionary
            *     N/A                      begin:1149    end:1154      Gold standard output tags 'virus' which
does
                                                                    not exist in dictionary
            *     N/a                      begin:1291    end:1296      Gold standard output tags 'virus' which
does
                                                                    not exist in dictionary
            *     N/A                      begin:703     end:708       Gold standard output tags 'virus' which
does
                                                                    not exist in dictionary
```

```
PMC7054940 Title:
       - Virus_SARS-CoV-2:
                  Tagger result             Gold standard output          Comment
              *     begin:9        end:26        begin:0        end:26        Gold standard output tags 'Emerging
novel
                                                                       coronavirus' which does not exist in dictionary.
                                                                       Tagger tags 'novel coronavirus'
              *     begin:28       end:37        begin:28       end:37         '2019-nCoV'


PMC7077245 Abstract:
       - Virus_other
              Tagger does not support this word class
       - Virus_SARS-CoV-2:
                  Tagger result             Gold standard output          Comment
              *     N/A                      begin:255      end:296       Gold standard output tags 'most recent
emergent
                                                                       group 2B coronavirus' which does not exist in
                                                                       dictionary
              *     begin:19       end:26        begin:19       end:36         'novel coronavirus'
              *     begin:38       end:47        begin:38       end:47         '2019-nCoV'

PMC7077245 Title:
       - Virus_SARS-CoV-2:
                  Tagger result             Gold standard output          Comment
              *     begin:14       end:25        begin:14       end:25         '2019-nCoV'
       - Virus_family
              Tagger does not support this word class

PMC7094943 Abstract:
       - Disease_other
              Tagger does not support this word class
       - Symptom_COVID-19:
                  Tagger result             Gold standard output          Comment
              *     begin:787      end:792       begin:787      end:792         'fever'
              *     begin:794      end:803       begin:794      end:803         'dizziness'
              *     begin:810      end:815       begin:810      end:815         'cough'
       - Virus_SARS-CoV-2:
                  Tagger result             Gold standard output          Comment
              *     N/A                      begin:996      end:1020      Gold standard output tags 'WH-Human 1'
which
                                                                       does not exist in dictionary
              *     begin:1056     end:1065      begin:1056     end:1065       '2019-nCoV'
              *     N/A                      begin:1159     end:1164      Gold standard output tags 'virus' which
does
                                                                       not exist in dictionary

       - Virus_family
              Tagger does not support this word class
```

```
PMC7094943 Title:
    - Virus_SARS-CoV-2:
            Tagger result              Gold standard output         Comment
        *   begin:2      end:17       begin:2      end:17        'new coronavirus'
    - Disease_other
        Tagger does not support this word class

PMC7110798 Abstract:
    - Disease_COVID-19:
            Tagger result              Gold standard output         Comment
        *   N/A                     begin:38      end:77        Gold standard output tags 'novel
coronavirus
                                                         (2019-nCoV) pneumonia' which does not exist
                                                         in dictionary
    - Virus_SARS-CoV-2:
            Tagger result              Gold standard output         Comment
        *   begin:38      end:55      N/A                     Due to priority rule longest match between
        *   begin:57      end:60      N/A                     Disease_COVID-19 and
Virus_SARS-CoV-2
                                                         the words aren't tagged, but since the word
                                                         is missing in the dictionary the tagger
                                                         tags the shorter words in Virus_SARS-CoV-2
                                                         dictionary.
        *   begin:267      end:276      begin:257      end:276       '2019-nCoV'
        *   begin:427      end:436      begin:427      end:436       '2019-nCoV'
        *   begin:1105     end:1114     begin:1105     end:1114      '2019-nCoV'
        *   begin:1216     end:1225     begin:1216     end:1225      '2019-nCoV'
    - Disease_other
        Tagger does not support this word class

PMC7110798 Title:
    - Virus_SARS-CoV-2:
            Tagger result              Gold standard output         Comment
        *   begin:59      end:76      begin:59      end:76        'novel coronavirus'
        *   begin:78      end:87      begin:78      end:87        '2019-nCoV'

PMC7159299 Abstract:
    - Symptom_COVID-19:
            Tagger result              Gold standard output         Comment
        *   begin:32      end:41      begin:32      end:41        'pneumonia'
        *   begin:1481     end:1486     begin:1481     end:1486      'fever'
        *   begin:1514     end:1519     begin:1514     end:1519      'cough'
        *   begin:1536     end:1543     begin:1536     end:1543      'myalgia'
        *   begin:1547     end:1554     begin:1547     end:1554      'fatigue'
        *   begin:1593     end:1610     begin:1593     end:1610      'sputum'
        *   begin:1629     end:1637     begin:1629     end:1637      'headache'
        *   N/A                     begin:1658     end:1669      Gold standard output tags 'haemoptysis'
which does
                                                         not exist in dictionary
        *   N/A                     begin:1692     end:1701      Gold standard output tags 'diarrhoea'
```