Problem 2

For this problem set, we will use
https://app.sketchengine.eu/#dashboard?corpname=preloaded%2Fcovid19

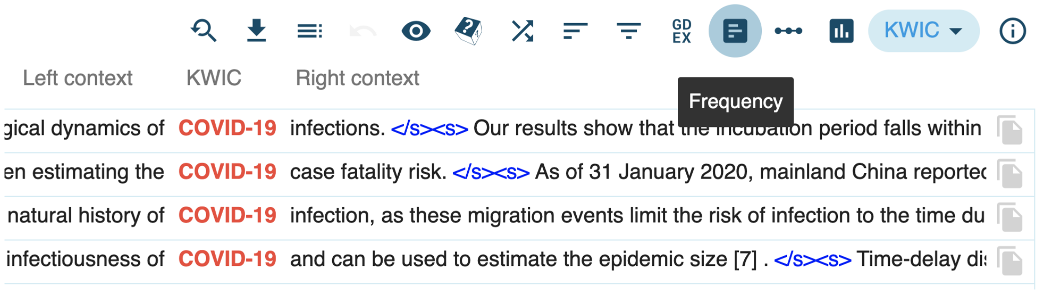
Problem 2.1

Click the above link, and follow this: Dashboard -> Concordance -> Advanced -> CQL.

Now write a query to find sentences containing all forms of covid and execute it. Some forms include covid-19, covid19, COVID19, covid-36, covid-54.

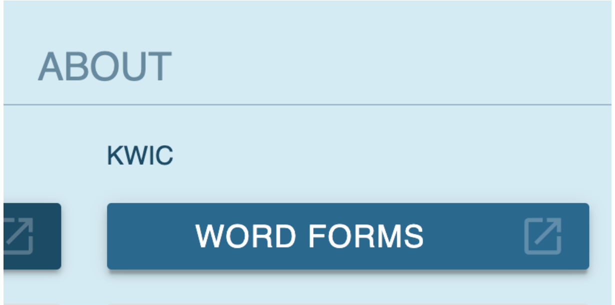
Once you get the sentences, click `Frequency -> KWIC > WORD FORMS` to generate the frequency of words. These steps are shown below:

Step 1:

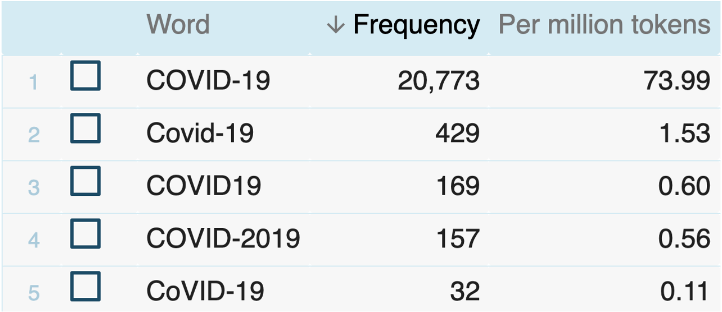


Left context	KWIC	Right context
gical dynamics of	COVID-19	infections. </s><s> Our results show that the incubation period falls within
en estimating the	COVID-19	case fatality risk. </s><s> As of 31 January 2020, mainland China reportec
natural history of	COVID-19	infection, as these migration events limit the risk of infection to the time du
infectiousness of	COVID-19	and can be used to estimate the epidemic size [7] . </s><s> Time-delay di

Step 2:



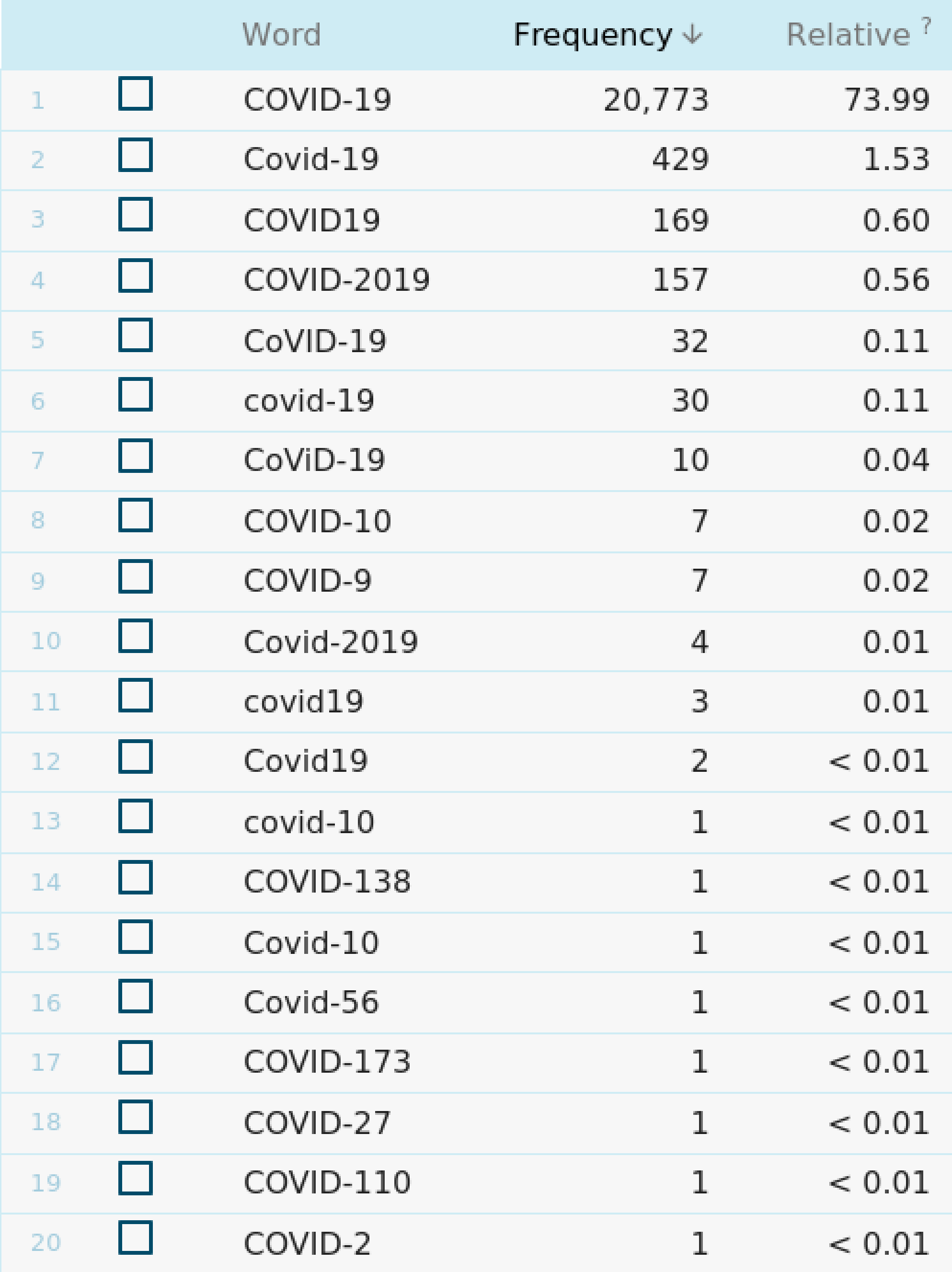
Step 3: The word list looks something like this:



		Word	↓ Frequency	Per million tokens
1	□	COVID-19	20,773	73.99
2	□	Covid-19	429	1.53
3	□	COVID19	169	0.60
4	□	COVID-2019	157	0.56
5	□	CoVID-19	32	0.11

What is the CQL query that you used for getting all forms of covid?
Answer: [word="(?i)covid-?\d+$"]

Include the snapshot of the top 20 words (5 words are shown above)?
Answer:



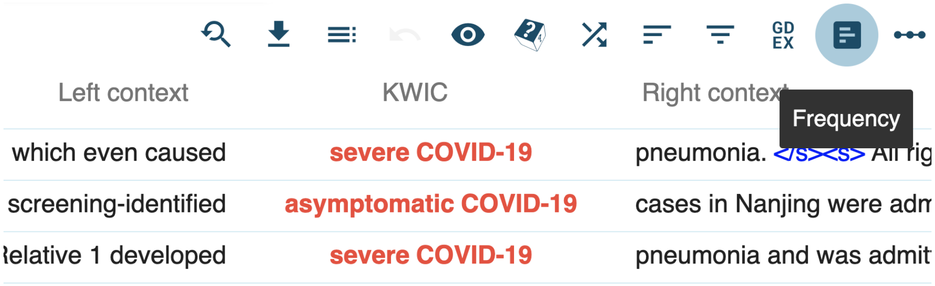
		Word	Frequency ↓	Relative ?
1	□	COVID-19	20,773	73.99
2	□	Covid-19	429	1.53
3	□	COVID19	169	0.60
4	□	COVID-2019	157	0.56
5	□	CoVID-19	32	0.11
6	□	covid-19	30	0.11
7	□	CoViD-19	10	0.04
8	□	COVID-10	7	0.02
9	□	COVID-9	7	0.02
10	□	Covid-2019	4	0.01
11	□	covid19	3	0.01
12	□	Covid19	2	< 0.01
13	□	covid-10	1	< 0.01
14	□	COVID-138	1	< 0.01
15	□	Covid-10	1	< 0.01
16	□	Covid-56	1	< 0.01
17	□	COVID-173	1	< 0.01
18	□	COVID-27	1	< 0.01
19	□	COVID-110	1	< 0.01
20	□	COVID-2	1	< 0.01

Problem 2.2

Let’s write CQL queries to find interesting words that occur in specific syntactic relations with covid (all forms). We did similar things in class. You will have to use tag and lemma in CQL queries. This tagset could be useful

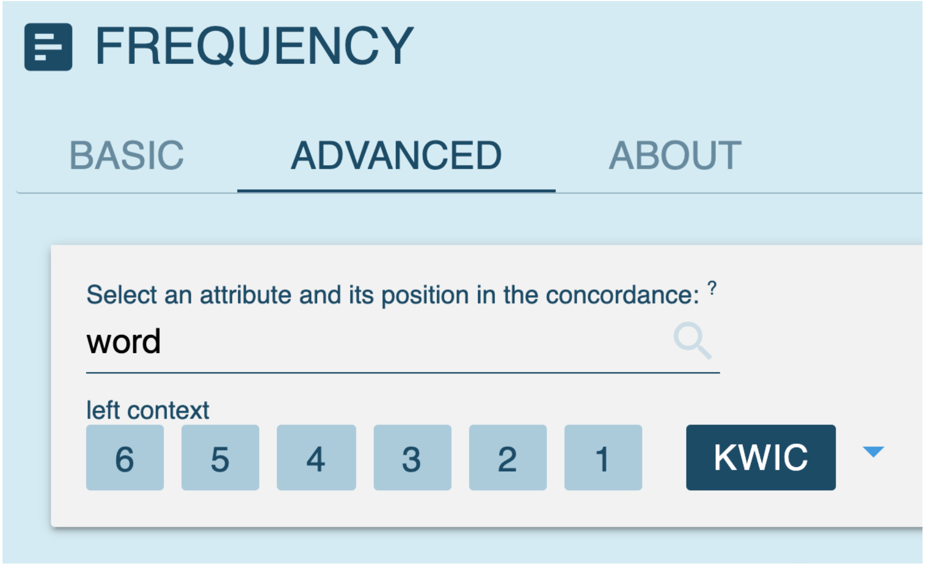
I will demonstrate how to get the modifiers of covid:

Step 1: First write a CQL query that produces concordance (examples) like this:



Left context	KWIC	Right context
which even caused	severe COVID-19	pneumonia. </s><s> All rig
screening-identified	asymptomatic COVID-19	cases in Nanjing were adm
Relative 1 developed	severe COVID-19	pneumonia and was admit

Step 2:

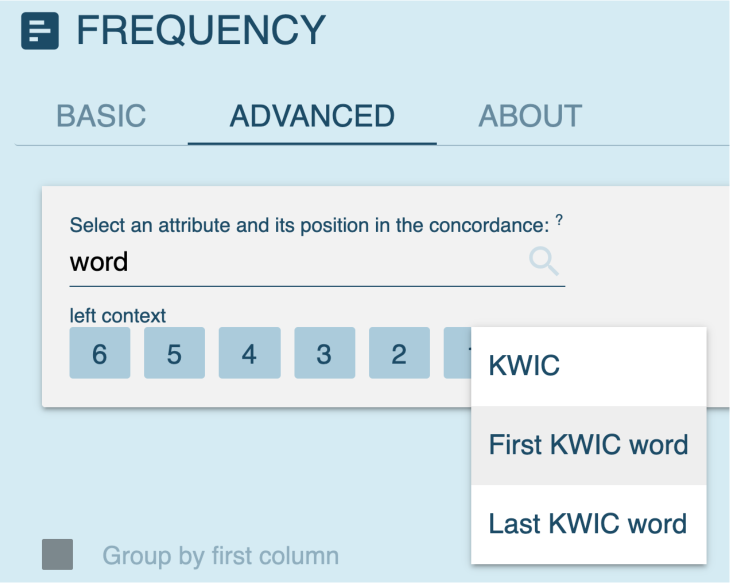


Select an attribute and its position in the concordance: ?

word

left context

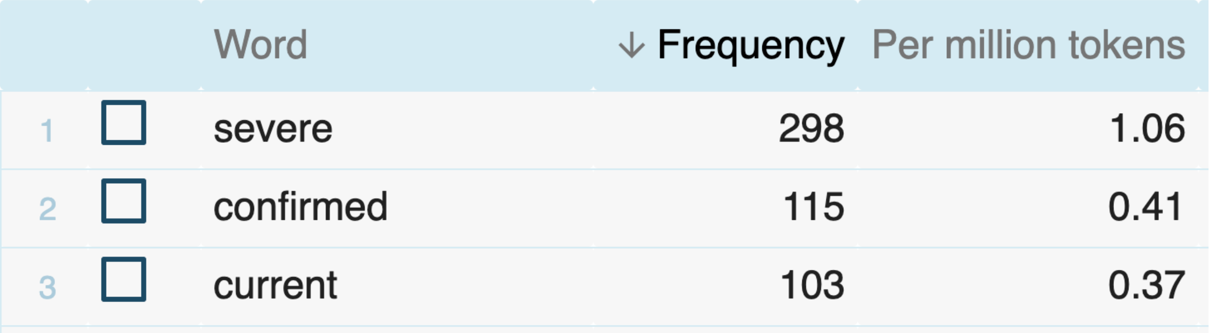
Step 3:



Select an attribute and its position in the concordance: ?
word

■ Group by first column

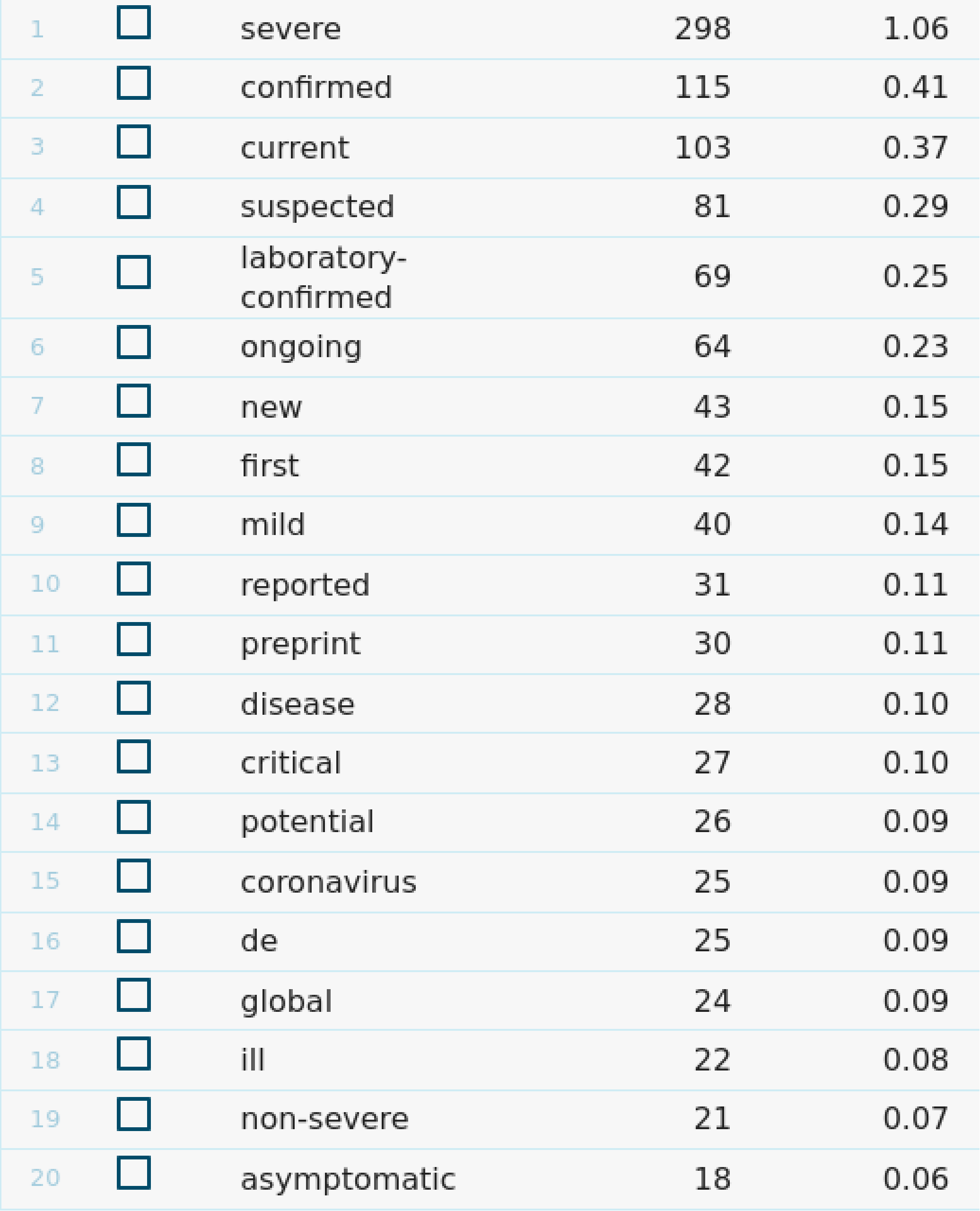
Step 4:



		Word	↓ Frequency	Per million tokens
1	□	severe	298	1.06
2	□	confirmed	115	0.41
3	□	current	103	0.37

What is the CQL query for modifiers of covid (all forms)?
Answer: [tag="J.*" | tag=”N.*”][word="(?i)covid-?\d+$"]

Include the snapshot of modifiers (top three are shown above)

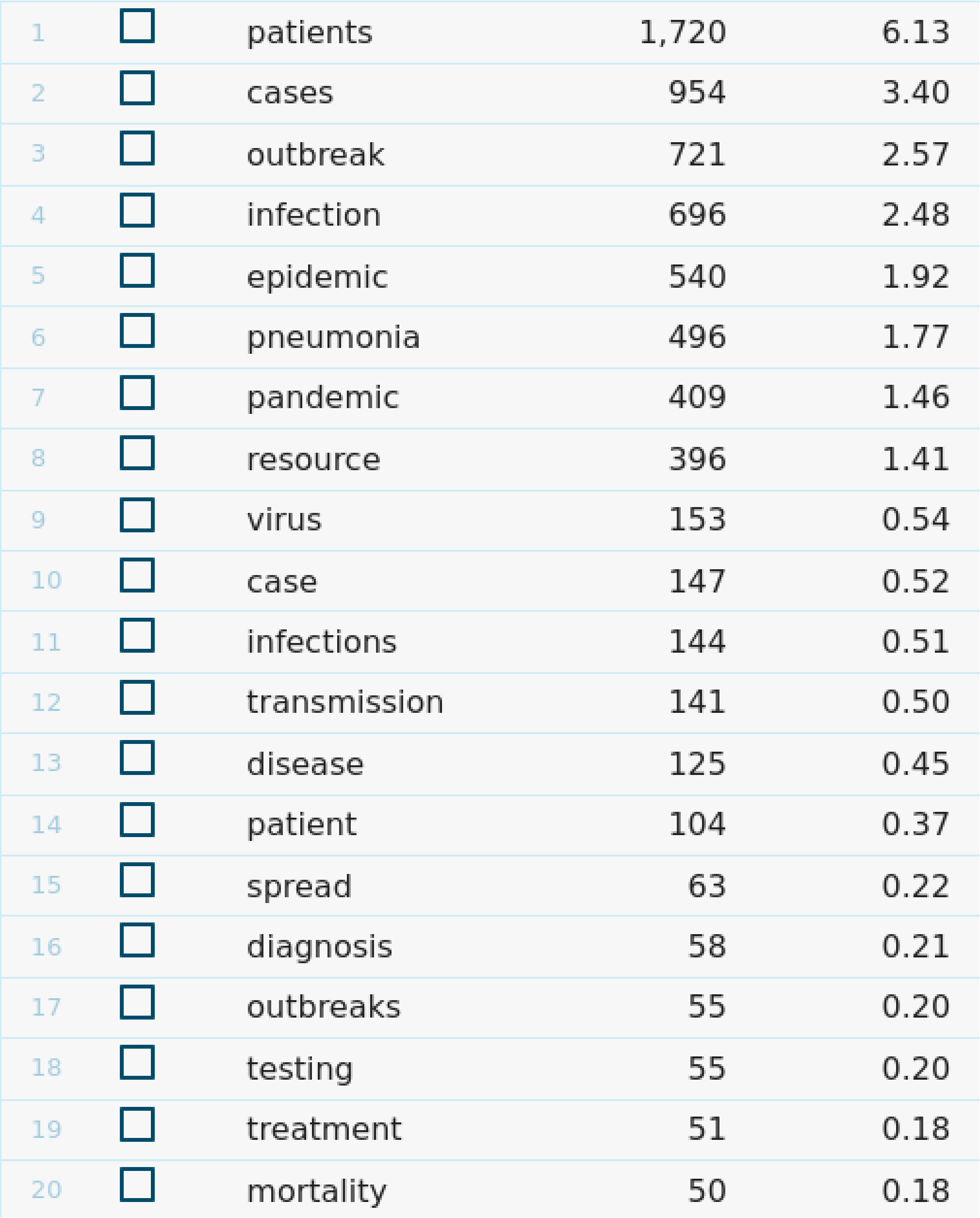


1	□	severe	298	1.06
2	□	confirmed	115	0.41
3	□	current	103	0.37
4	□	suspected	81	0.29
5	□	laboratory-confirmed	69	0.25
6	□	ongoing	64	0.23
7	□	new	43	0.15
8	□	first	42	0.15
9	□	mild	40	0.14
10	□	reported	31	0.11
11	□	preprint	30	0.11
12	□	disease	28	0.10
13	□	critical	27	0.10
14	□	potential	26	0.09
15	□	coronavirus	25	0.09
16	□	de	25	0.09
17	□	global	24	0.09
18	□	ill	22	0.08
19	□	non-severe	21	0.07
20	□	asymptomatic	18	0.06

What is the CQL query of words that are modified by covid (all forms)?

Answer: [word="(?i)covid-?\d+$"][tag="N.*"]

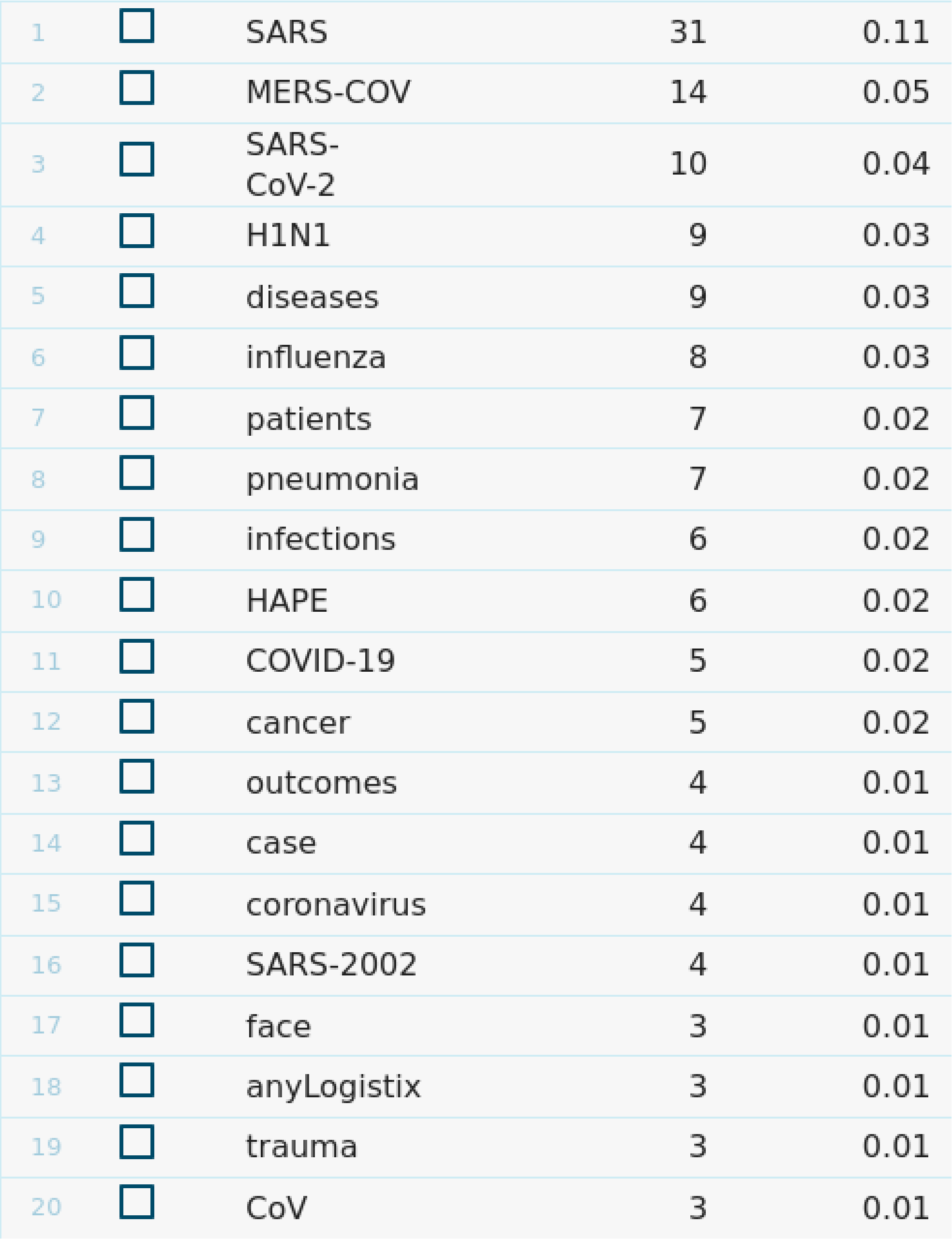
Include the snapshot of those words



1	□	patients	1,720	6.13
2	□	cases	954	3.40
3	□	outbreak	721	2.57
4	□	infection	696	2.48
5	□	epidemic	540	1.92
6	□	pneumonia	496	1.77
7	□	pandemic	409	1.46
8	□	resource	396	1.41
9	□	virus	153	0.54
10	□	case	147	0.52
11	□	infections	144	0.51
12	□	transmission	141	0.50
13	□	disease	125	0.45
14	□	patient	104	0.37
15	□	spread	63	0.22
16	□	diagnosis	58	0.21
17	□	outbreaks	55	0.20
18	□	testing	55	0.20
19	□	treatment	51	0.18
20	□	mortality	50	0.18

What is the CQL query for words that occur in right coordination with covid (all forms) (e.g., in COVID-19 , SARS-2002 , and HCoV-NL63, the words iSARS-2002 and HCoV-NL63 are the right conjucts/coordinates).
Answer: [word="(?i)covid-?\d+$"]([word=","]?[tag="CC"][tag="J.*"]{0,3}[tag="N.*"]){1,10}

Include the snapshot of those words

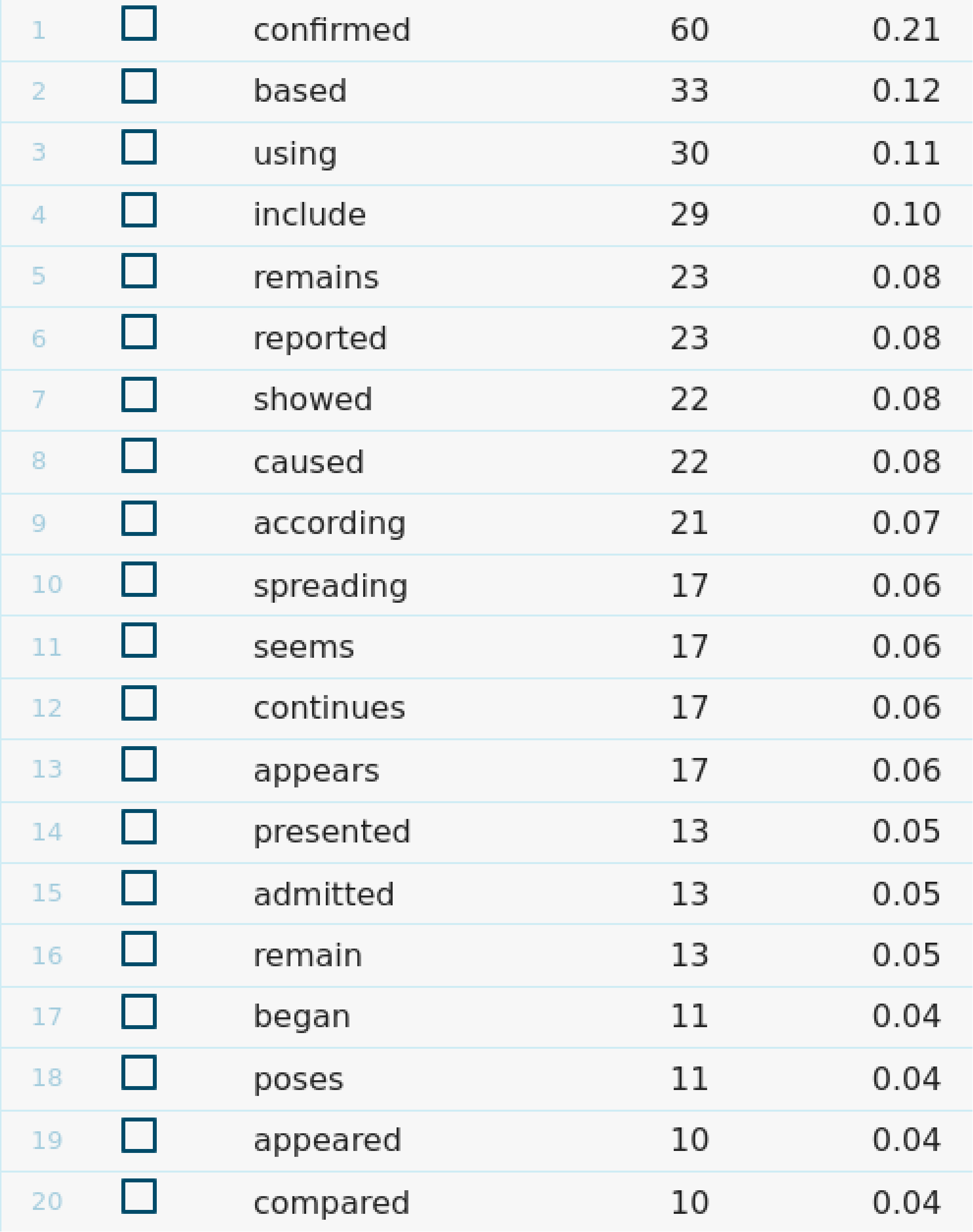


1	□	SARS	31	0.11
2	□	MERS-COV	14	0.05
3	□	SARS-CoV-2	10	0.04
4	□	H1N1	9	0.03
5	□	diseases	9	0.03
6	□	influenza	8	0.03
7	□	patients	7	0.02
8	□	pneumonia	7	0.02
9	□	infections	6	0.02
10	□	HAPE	6	0.02
11	□	COVID-19	5	0.02
12	□	cancer	5	0.02
13	□	outcomes	4	0.01
14	□	case	4	0.01
15	□	coronavirus	4	0.01
16	□	SARS-2002	4	0.01
17	□	face	3	0.01
18	□	anyLogistix	3	0.01
19	□	trauma	3	0.01
20	□	CoV	3	0.01

What is the CQL query for verbs that can take covid (all forms) as subject?

Answer: [word="(?i)covid-?\d+$"][tag="V.*" & lemma!="(be|have|do)"]
Filtering out main auxiliary verbs

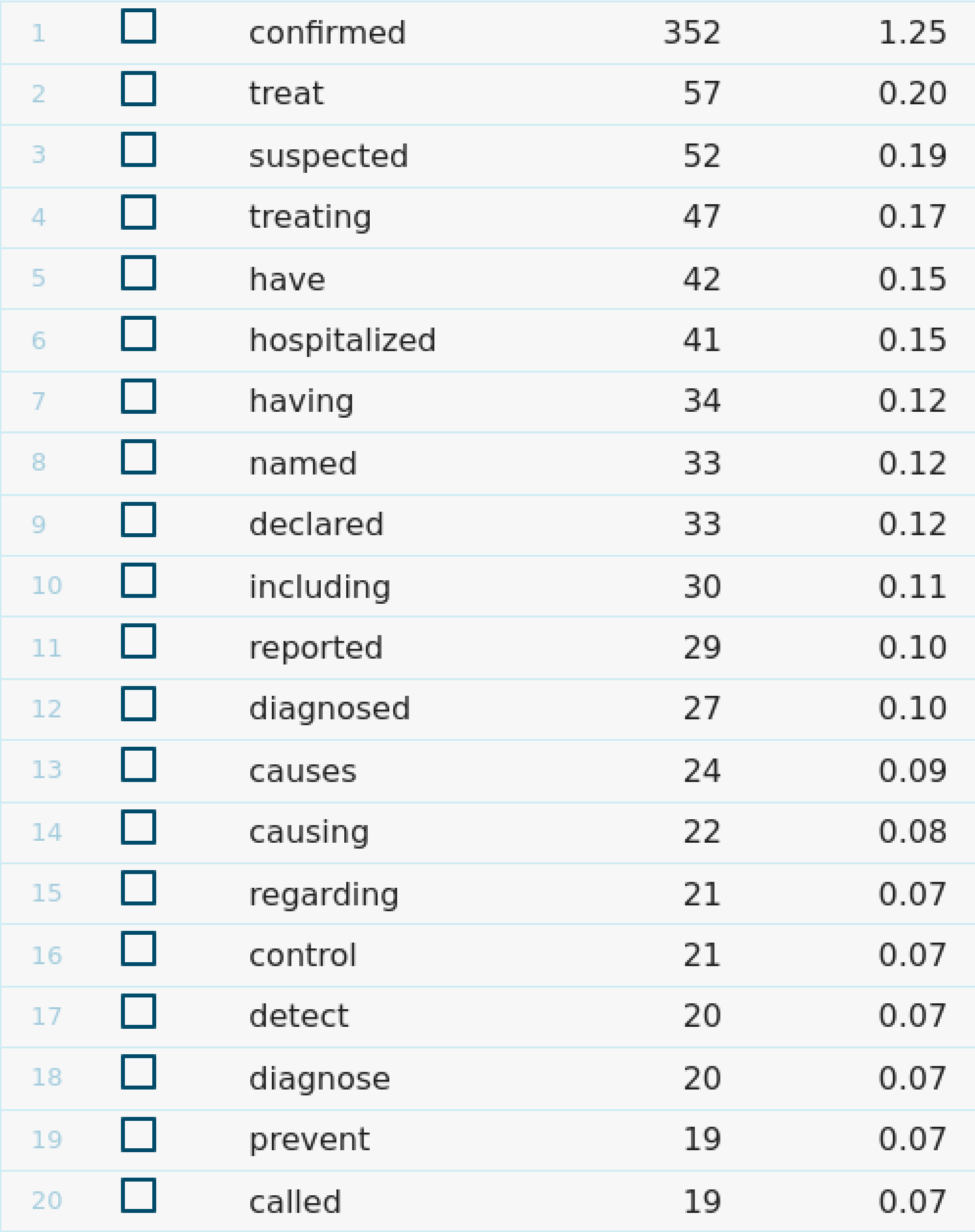
Include the snapshot of verbs that take covid as subject



1	□	confirmed	60	0.21
2	□	based	33	0.12
3	□	using	30	0.11
4	□	include	29	0.10
5	□	remains	23	0.08
6	□	reported	23	0.08
7	□	showed	22	0.08
8	□	caused	22	0.08
9	□	according	21	0.07
10	□	spreading	17	0.06
11	□	seems	17	0.06
12	□	continues	17	0.06
13	□	appears	17	0.06
14	□	presented	13	0.05
15	□	admitted	13	0.05
16	□	remain	13	0.05
17	□	began	11	0.04
18	□	poses	11	0.04
19	□	appeared	10	0.04
20	□	compared	10	0.04

What is the CQL query for verbs that can take covid (all forms) as object?
Answer: [tag="V.*"][tag="J.*"]{0,3}[word="(?i)covid-?\d+$"]

Include the snapshot of verbs that take COVID as object.



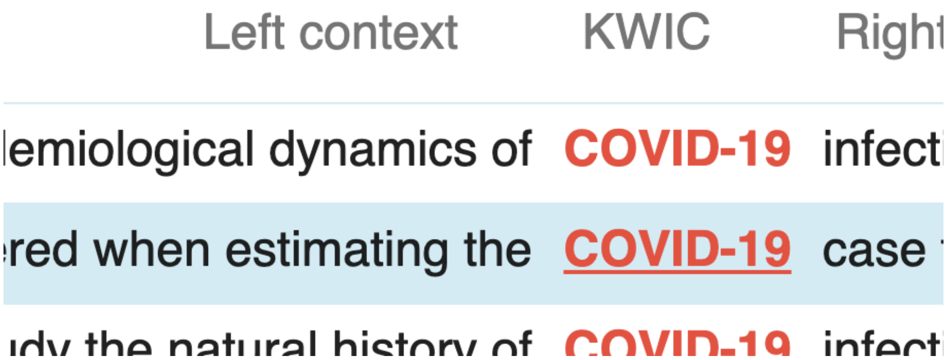
1	□	confirmed	352	1.25
2	□	treat	57	0.20
3	□	suspected	52	0.19
4	□	treating	47	0.17
5	□	have	42	0.15
6	□	hospitalized	41	0.15
7	□	having	34	0.12
8	□	named	33	0.12
9	□	declared	33	0.12
10	□	including	30	0.11
11	□	reported	29	0.10
12	□	diagnosed	27	0.10
13	□	causes	24	0.09
14	□	causing	22	0.08
15	□	regarding	21	0.07
16	□	control	21	0.07
17	□	detect	20	0.07
18	□	diagnose	20	0.07
19	□	prevent	19	0.07
20	□	called	19	0.07

Problem 2.3

What are the most important words that form collocations with COVID (where covid is the right word)?

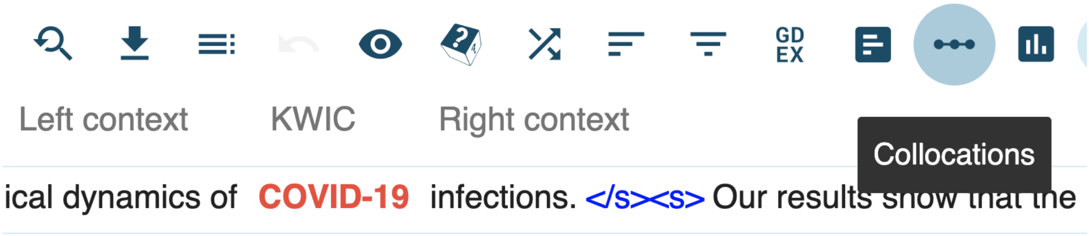
You can generate collocations as follows: First get concordance of all forms of covid.

Step 1:



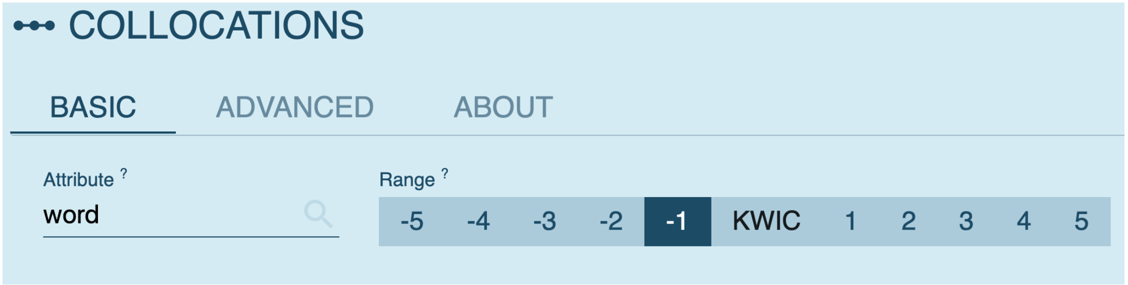
Left context	KWIC	Righ
lemiological dynamics of	COVID-19	infect
ıred when estimating the	COVID-19	case
ıdy the natural history of	COVID-19	infect

Step 2:



Left context	KWIC	Right context
ical dynamics of	COVID-19	infections. </s><s> Our results show that the

Step 3:



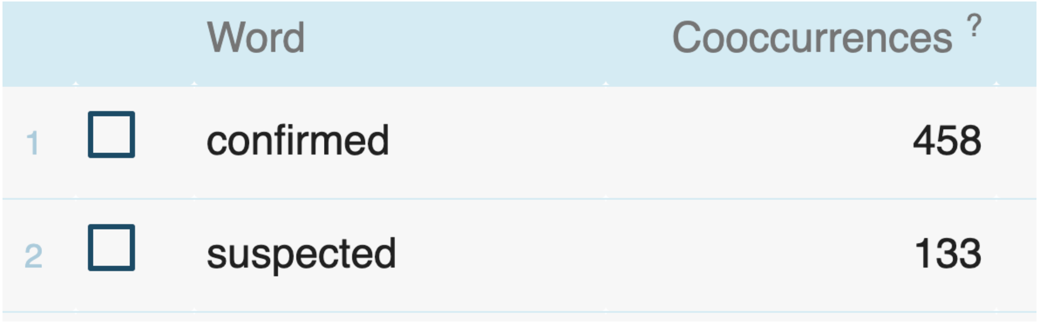
••• COLLOCATIONS

BASIC ADVANCED ABOUT

Attribute ? word

Range ? -5 -4 -3 -2 **-1** KWIC 1 2 3 4 5

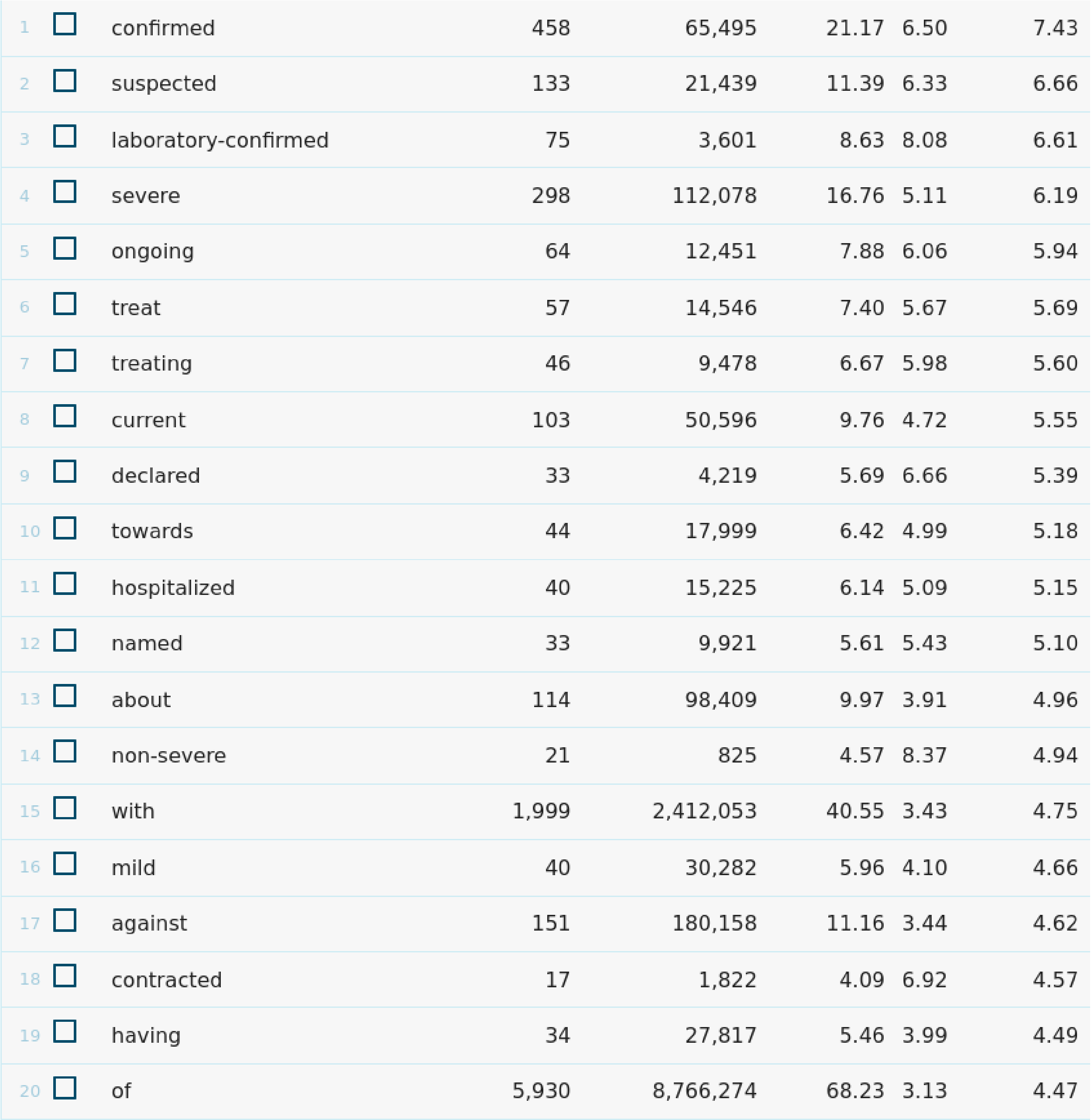
Step 4:



		Word	Cooccurrences ?
1	□	confirmed	458
2	□	suspected	133

Show the collocations sorted according to what you think is the best metric (T-Score, MI, LogDice). Indicate the metric you used.

LogDice appears to be best at finding appropriate collocations:



1	□	confirmed	458	65,495	21.17	6.50	7.43
2	□	suspected	133	21,439	11.39	6.33	6.66
3	□	laboratory-confirmed	75	3,601	8.63	8.08	6.61
4	□	severe	298	112,078	16.76	5.11	6.19
5	□	ongoing	64	12,451	7.88	6.06	5.94
6	□	treat	57	14,546	7.40	5.67	5.69
7	□	treating	46	9,478	6.67	5.98	5.60
8	□	current	103	50,596	9.76	4.72	5.55
9	□	declared	33	4,219	5.69	6.66	5.39
10	□	towards	44	17,999	6.42	4.99	5.18
11	□	hospitalized	40	15,225	6.14	5.09	5.15
12	□	named	33	9,921	5.61	5.43	5.10
13	□	about	114	98,409	9.97	3.91	4.96
14	□	non-severe	21	825	4.57	8.37	4.94
15	□	with	1,999	2,412,053	40.55	3.43	4.75
16	□	mild	40	30,282	5.96	4.10	4.66
17	□	against	151	180,158	11.16	3.44	4.62
18	□	contracted	17	1,822	4.09	6.92	4.57
19	□	having	34	27,817	5.46	3.99	4.49
20	□	of	5,930	8,766,274	68.23	3.13	4.47

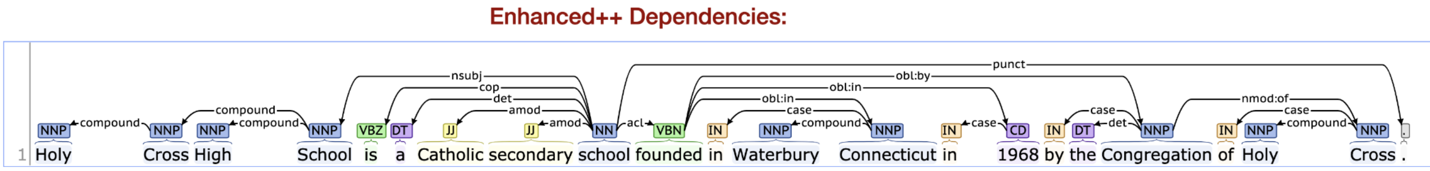
Problem 3:

Write SemGrex regular expressions that can detect organizations and their founders. Make use of https://corenlp.run to parse sentences to syntactic graphs and for running SemGrex expressions.

Here is an example:

Holy Cross High School is a Catholic secondary school founded in Waterbury Connecticut in 1968 by the **Congregation of Holy Cross .**

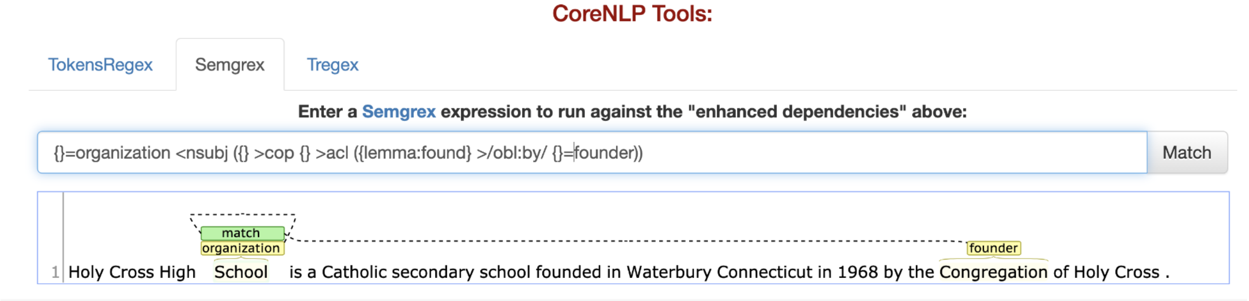
The corresponding Enhanced++ Dependencies syntactic graph is as follows:



Enhanced++ Dependencies:

The below SemGrex pattern extracts the headword of the organization and the headword of the founder.

{}=organization <nsubj ({} >acl ({lemma:found} >/obl:by/ {}=founder))



CoreNLP Tools:

Enter a Semgrex expression to run against the "enhanced dependencies" above:

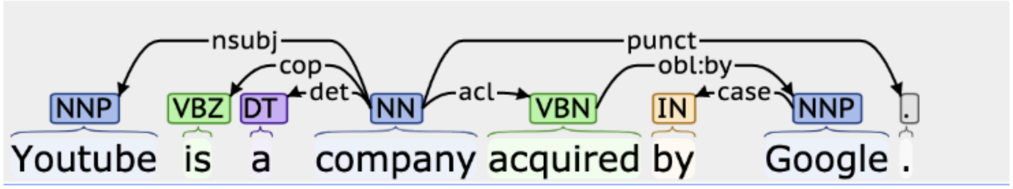
{}=organization <nsubj ({} >cop {} >acl ({lemma:found} >/obl:by/ {}=founder))

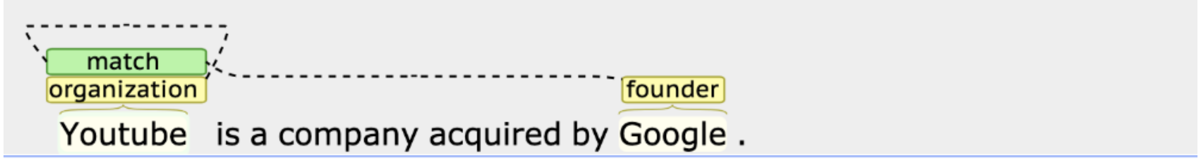
1 Holy Cross High School is a Catholic secondary school founded in Waterbury Connecticut in 1968 by the Congregation of Holy Cross .

This pattern can be read as the “organization” that is a subject of something, and this something is founded by the founder.

Here it extracts School (i.e., the headword of Holy Cross High School) as the organization and Congregation (i.e., the headword of the Congregation of Holy Cross) as the founder.

Your goal is to write SemGrex expressions that can generalize to multiple sentences but at the same time don’t match incorrect sentences. For example, if you don’t use {lemma:found} in the above sentence, your pattern will also match a sentence like “Youtube is a company acquired by Google” (see below.)



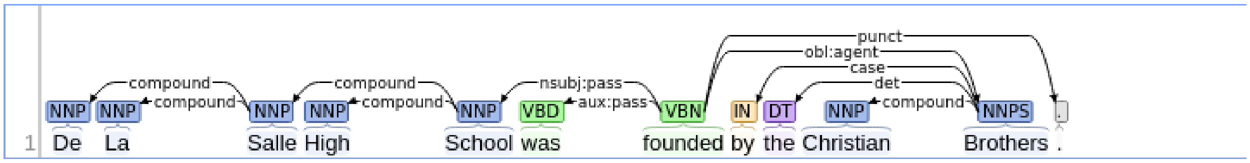


Problem 3.1

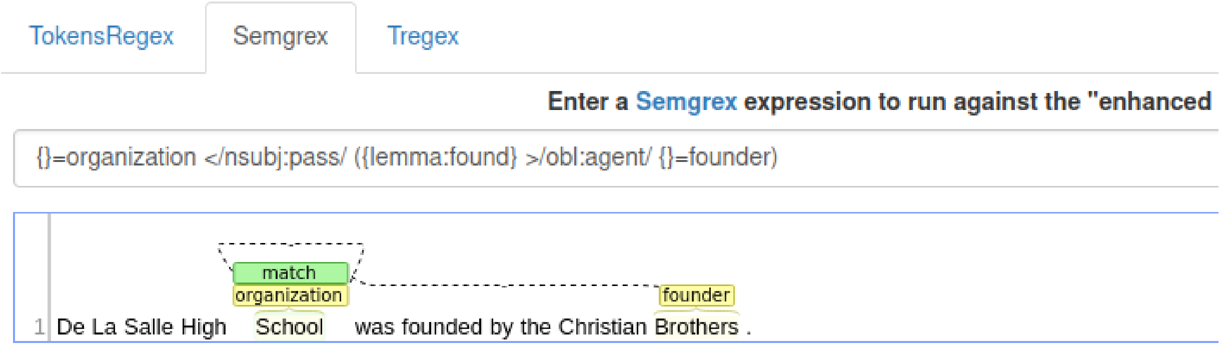
Write the SemGrex patterns for the following sentences that extract the organization name (headword is enough) and its founder (headword is enough). Sentences that can make use of the same expression should be in the same snapshot (containing Enhanced++ Dependencies, Semgrex expression, and the matchings):

De La Salle High School was founded by **the Christian Brothers** .

Enhanced++ Dependencies:


punct
obl:agent
case
det
compound
compound
compound
compound
compound
nsubj:pass
aux:pass
compound
NNP NNP NNP NNP NNP VBD VBN IN DT NNP NNPS .
1 De La Salle High School was founded by the Christian Brothers .

CoreNLP Tools:


TokensRegex
Semgrex
Tregex

Enter a Semgrex expression to run against the "enhanced

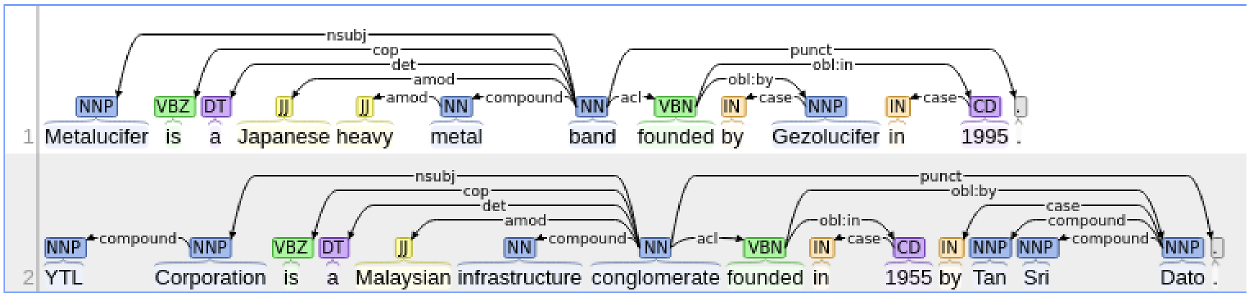
```
{}=organization </nsubj:pass/ ({lemma:found} >/obl:agent/ {}=founder)
```

match
organization
founder
1 De La Salle High School was founded by the Christian Brothers .

Metalucifer is a Japanese heavy metal band founded by **Gezolucifer** in 1995 .

YTL Corporation is a Malaysian infrastructure conglomerate founded in 1955 by **Tan Sri Dato**.

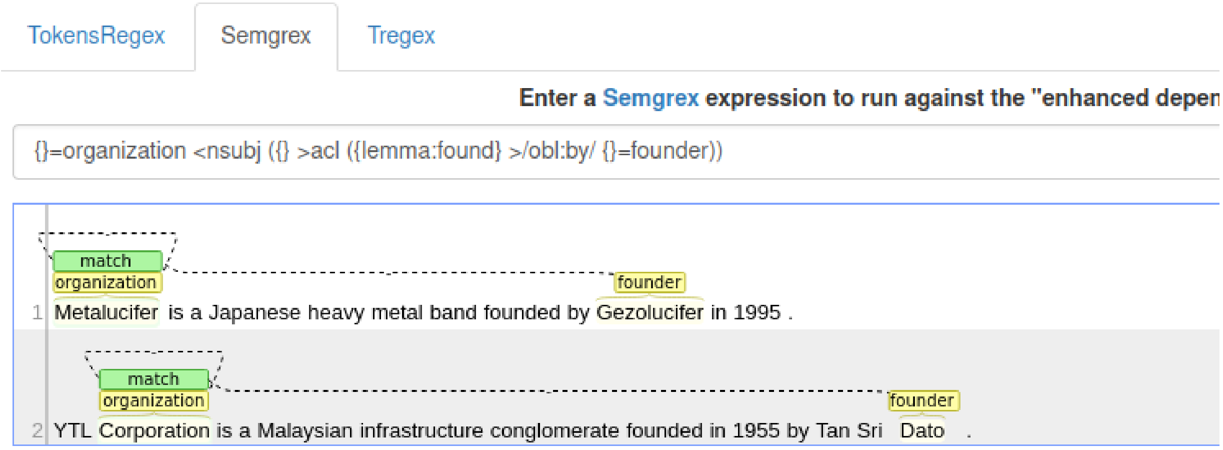
Enhanced++ Dependencies:


nsubj
cop
det
amod
compound
acl
obl:by
obl:in
case
punct
NNP VBZ DT JJ JJ NN NN VBN IN NNP IN CD .
NNP NNP VBZ DT JJ NN NN VBN IN CD IN NNP NNP NNP .

1 Metalucifer is a Japanese heavy metal band founded by Gezolucifer in 1995 .

2 YTL Corporation is a Malaysian infrastructure conglomerate founded in 1955 by Tan Sri Dato .

CoreNLP Tools:



Enter a Semgrex expression to run against the "enhanced depen

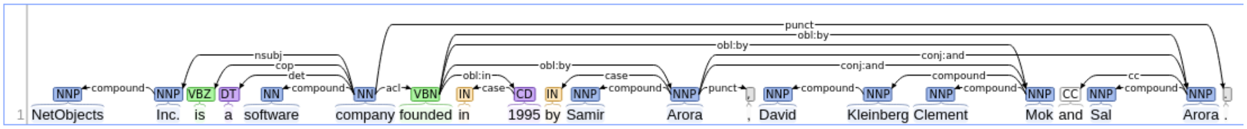
{}=organization <nsubj ({} >acl ({lemma:found} >/obl:by/ {}=founder))

1 Metalucifer is a Japanese heavy metal band founded by Gezolucifer in 1995 .

2 YTL Corporation is a Malaysian infrastructure conglomerate founded in 1955 by Tan Sri Dato .

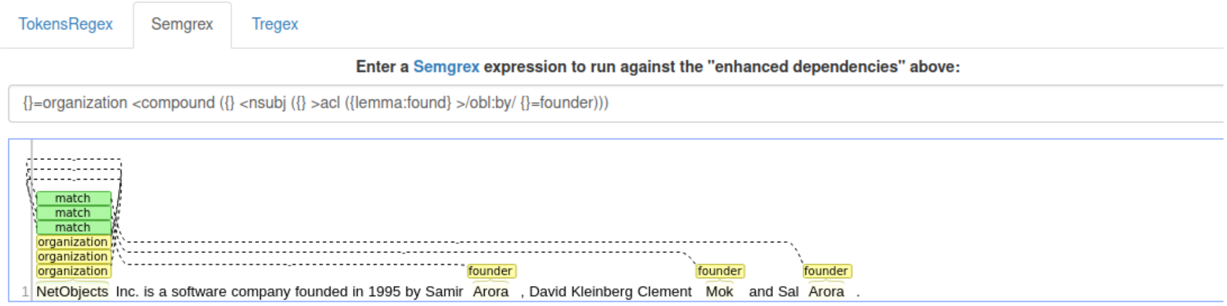
NetObjects Inc. is a software company founded in 1995 by **Samir Arora, David Kleinberg Clement Mok** and **Sal Arora .**
(If there are multiple founders, you have to extract headword corresponding to each founder)

Enhanced++ Dependencies:


compound
nsubj
cop
det
acl
obl:in
obl:by
case
punct
conj:and
cc
NNP NNP VBZ DT NN NN VBN IN CD IN NNP NNP . NNP NNP NNP NNP CC NNP NNP .

1 NetObjects Inc. is a software company founded in 1995 by Samir Arora , David Kleinberg Clement Mok and Sal Arora .

CoreNLP Tools:

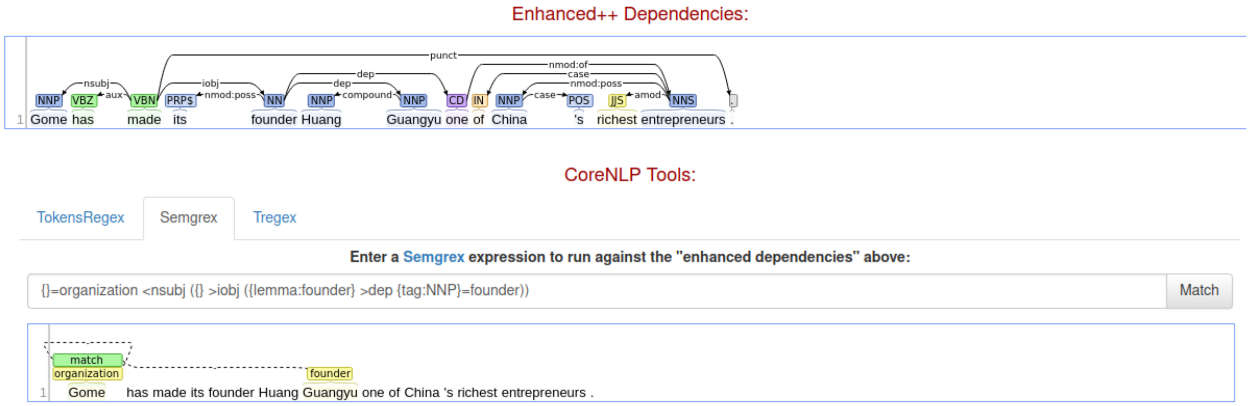


Enter a Semgrex expression to run against the "enhanced dependencies" above:

{}=organization <compound ({} <nsubj ({} >acl ({lemma:found} >/obl:by/ {}=founder)))

1 NetObjects Inc. is a software company founded in 1995 by Samir Arora , David Kleinberg Clement Mok and Sal Arora .

Gome has made its founder **Huang Guangyu** one of China's richest entrepreneurs.


Enhanced++ Dependencies:

NNP VBZ VBN PRP$ NN NNP NNP CD IN NNP POS JJS NNS .
nsubj
aux
iobj
nmod:poss
dep
dep
compound
punct
nmod:of
case
nmod:poss
case
amod
1 Gome has made its founder Huang Guangyu one of China 's richest entrepreneurs .

CoreNLP Tools:

TokensRegex
Semgrex
Tregex

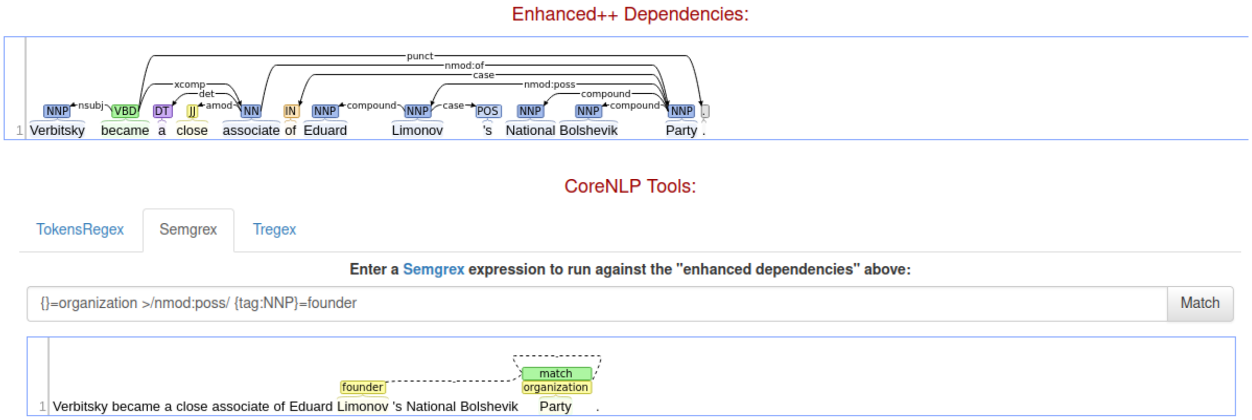
Enter a Semgrex expression to run against the "enhanced dependencies" above:

```
{}=organization <nsubj ({} >iobj ({lemma:founder} >dep {tag:NNP}=founder))
```

Match
match
organization
founder

1 Gome has made its founder Huang Guangyu one of China 's richest entrepreneurs .

Verbitsky became a close associate of **Eduard Limonov**'s **National Bolshevik Party**.


Enhanced++ Dependencies:

NNP VBD DT JJ NN IN NNP NNP POS NNP NNP NNP .
nsubj
xcomp
det
amod
punct
nmod:of
case
dep
compound
case
nmod:poss
compound
compound
1 Verbitsky became a close associate of Eduard Limonov 's National Bolshevik Party .

CoreNLP Tools:

TokensRegex
Semgrex
Tregex

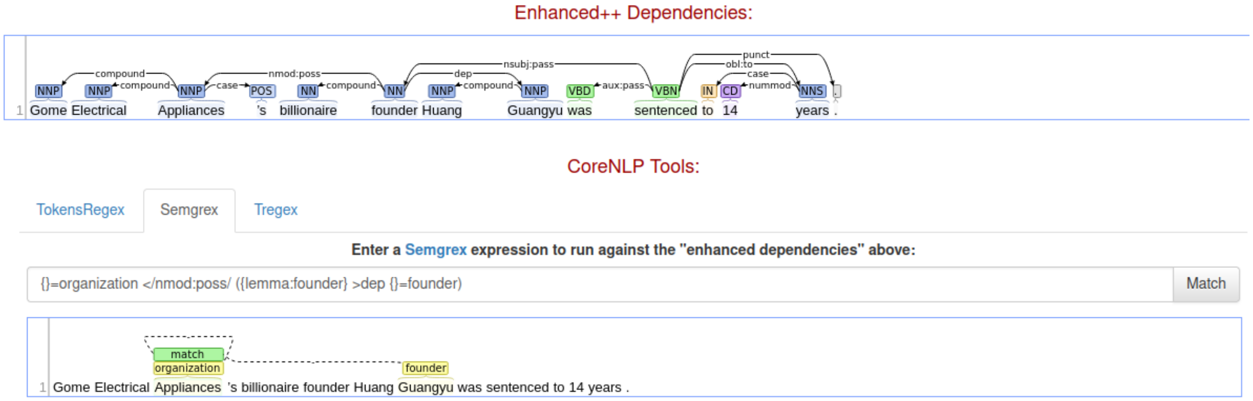
Enter a Semgrex expression to run against the "enhanced dependencies" above:

```
{}=organization >/nmod:poss/ {tag:NNP}=founder
```

Match
founder
match
organization

1 Verbitsky became a close associate of Eduard Limonov 's National Bolshevik Party .

Gome Electrical Appliances’s billionaire founder **Huang Guangyu** was sentenced to 14 years.


Enhanced++ Dependencies:

NNP NNP NNP POS NN NN NNP NNP VBD VBN IN CD NNS .
compound
compound
case
nmod:poss
compound
dep
compound
nsubj:pass
aux:pass
punct
obl:to
case
nummod
1 Gome Electrical Appliances 's billionaire founder Huang Guangyu was sentenced to 14 years .

CoreNLP Tools:

TokensRegex
Semgrex
Tregex

Enter a Semgrex expression to run against the "enhanced dependencies" above:

```
{}=organization </nmod:poss/ ({lemma:founder} >dep {}=founder)
```

Match
match
organization
founder

1 Gome Electrical Appliances 's billionaire founder Huang Guangyu was sentenced to 14 years .