

Автономная некоммерческая общеобразовательная организация "Физтех-лицей"

(АНОО «Физтех-лицей» им. П.Л. Капицы)

XX научно-практическая конференция

«Старт в инновации»

**Применение нейросетевых методов анализа текста
и машинного обучения для извлечения закономерностей
из корпуса научных статей о COVID-19**

Выполнила:

Сошникова Виктория, 10 класс

Руководитель:

Сошников Дмитрий Валерьевич

Московская область, г. Долгопрудный

2021 г.

Краткое содержание

В конце 2019 года все человечество столкнулось с колоссальной проблемой - новой коронавирусной инфекцией COVID-19. Когда стало понятно, что нужно в кратчайшие сроки начинать бороться с вирусом, тысячи врачей по всему миру стали проводить исследования в различных областях медицины. Количество статей по данной теме резко возросло, поэтому усложнилась задача поиска и обработки нужной информации. Мы решили помочь исследователям и создать инструменты, которые помогут систематизировать и упорядочить большое количество данных, выделив из статей медицинские сущности и другую полезную информацию. Для наглядной визуализации мы построим несколько типов графиков, наглядно показывающих взаимосвязи среди терминов.

Введение

В настоящее время поток информации, производимой человеком, невообразимо велик. Многие современные ученые делятся своими знаниями и открытиями посредством публикации научных статей, но, как нам кажется, из-за огромного количества новой хаотичной информации, становится все сложнее найти интересующие статьи по конкретным темам. Например, если человек хочет найти статьи по теме поиска лекарства от коронавируса, ему придется вручную искать нужную тему в тысячах статей, читать весь текст и собственноручно выделять нужную ему информацию. Мы хотим упростить эту задачу, тем самым ускорить изучение болезни и разработку способов борьбы с ней.

Датасет CORD-19

В качестве источника данных мы будем использовать датасет CORD-19[2], на данный момент содержащий более 400 000 статей, при этом 150 000 из них с полным текстом, о COVID-19 и других связанных с ним болезнях. Этот набор данных был создан Институтом Аллена по искусственному интеллекту в партнерстве с Инициативой Чан Цукерберга, Центром безопасности и новых технологий Джорджтаунского университета, а также Национальной медицинской библиотекой и IBM. Авторы призывают людей использовать данные для создания инструментов для обработки и анализа данных, чтобы помочь медицинскому сообществу. Сейчас доступны более 30 гигабайт статей, при этом датасет регулярно дополняется.

Выделение сущностей

Первым этапом работы необходимо обработать научные статьи и выделить из них медицинские сущности, то есть базовые понятия, например, такие как названия лекарств и диагнозов. Для этого используются нейросети.

В последние годы наблюдается существенный прогресс в области обработки естественного языка. Нейросети способны решать следующие задачи:

- Машинный перевод
- Ответы на вопросы
- Сбор новостей
- Категоризация текста
- Выделение сущностей

В 2018 году Джейкобом Девлином и его коллегами из Google была создана языковая модель BERT[3][4] (Bidirectional Encoder Representations from Transformers), а уже в 2019 году она показала лучшие результаты во всех областях обработки. В октябре 2020 года почти все запросы Google на английском языке были обработаны при помощи BERT.

Данная модель обучена на огромном количестве различных текстов из интернета. Для более специализированных задач в области биомедицины необходимо переобучить модель на научных публикациях. Существует несколько проектов для решения этой задачи [5][6]. Для своего проекта мы решили использовать PubMedBert[7], которую переобучили исследователи из Microsoft Research на основе модели BERT и базы данных медицинских и биологических статей PubMed[8]. Модель решает более общую задачу, чем нам нужно. Именно для выделения сущностей существует сервис Text Analytics for health, созданный компанией Microsoft. Для обработки данных у него есть следующие функции:

1. Выделение сущностей

Ribavirin UMLS: C0035525 was also evaluated against SARS-CoV-2 infection UMLS: C5203670, but the antiviral UMLS: C0003451 property of drugs UMLS: C0013227 is still not well established against the SARS-CoV-2 UMLS: C5203670 negation.

In addition, after oral UMLS: C0017189 administration, the drug was rapidly absorbed into the GI tract UMLS: C0017189.

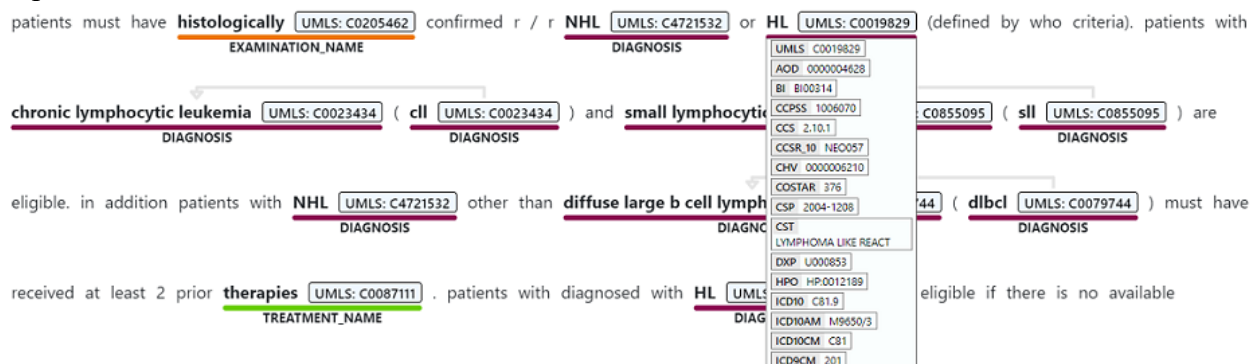
The drug has oral bioavailability UMLS: C0017189 around 64 UMLS: C0017189 % UMLS: C0017189 with large volume of distribution.

2. Выделение связей между сущностями

Woman UMLS: C0043210 in NAD UMLS: C2051415 with a h/o CAD UMLS: C1956346, DM2 UMLS: C0011860, asthma UMLS: C0004096 and HTN UMLS: C0020538 on ramipril UMLS: C0072973 for 8 years UMLS: C0072973 awoke from sleep around 2:30 am this morning UMLS: C0242429 of a sore throat UMLS: C0242429 and swelling of tongue UMLS: C0236068. She came immediately to the ED b/c she was having difficulty swallowing UMLS: C0011168 and some trouble breathing UMLS: C0013404 due to

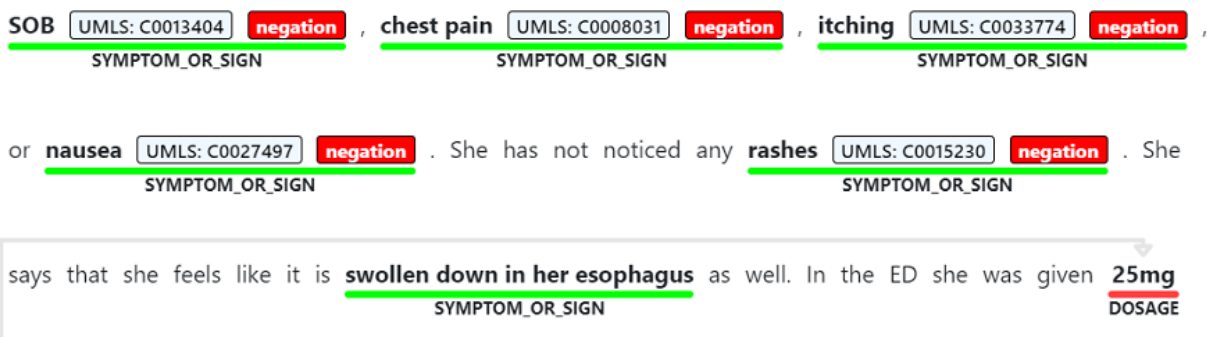
3. Привязка сущностей к медицинским онтологиям, то есть иерархическим системам представления терминов и их взаимосвязей. В дальнейшем мы будем использовать онтологию UMLS (Unified Medical Language System), это медицинская

онтология, которая используется для разработки компьютерных систем, понимающих медицинские термины.



4. Выявление отрицания сущностей

She has never had a similar reaction ever before and she did not have any associated



В качестве результата сервис возвращает Json файл, в котором хранится вся нужная нам информация о статье: название, дату публикации, имена авторов, сущности и связи между ними и другое. Далее с ней будет удобно работать за счет иерархической системы представления данных

Укороченный пример Json файла для одной статьи

```
{
  "id": "klg8rjpk",
  "title": "Vaccines and Airline Travel: A Federal Role to Protect the Public Health.",
  "authors": "Robertson, Christopher T",
  "journal": "American journal of law & medicine",
  "abstract": "This Article explores two ways in which airline travel is an important vector for the spread of infectious disease, and argues that airlines have market-based and liability-based reasons to require that passengers be vaccinated. Going further, the Article explores whether the federal government has the legal and constitutional authority-especially under the Commerce Clause-to encourage or mandate that airlines implement such a vaccine screen. By dis
```

rupting the spread of disease at key network nodes where individuals interact and then connect with other geographic regions, and by creating another incentive for adult vaccination, an airline vaccine screen could be an effective and legally viable tool for the protection of public health.",

```
"publish_time": "2016",
"entities": [
  {
    "offset": 96,
    "length": 18,
    "text": "infectious disease",
    "category": "Diagnosis",
    "confidenceScore": 0.99,
    "isNegated": false,
    "links": [
      {
        "dataSource": "UMLS",
        "id": "C0009450"
      },
      {
        "dataSource": "AOD",
        "id": "0000004780"
      },
      ...]...
    ]
  },
  ...]...
]

"relations": [
  {
    "relationType": "QualifierOfCondition",
    "bidirectional": false,
    "source": {
      "uri": "#/results/documents/1/entities/3",
      "text": "disrupting",
      "category": "ConditionQualifier",
      "isNegated": false,
      "offset": 450
    },
    "target": {
      "uri": "#/results/documents/1/entities/5",
      "text": "disease",
      "category": "Diagnosis",
      "isNegated": false,
      "offset": 475
    }
  },
  ...]...
]
```

Работать с такими данными можно в любом языке программирования, например, Python. Но для выполнения специализированных запросов придется разбивать файл на массивы списки и строки и писать не самую простую программу. Мы решили

воспользоваться готовой системой управления базами данных Cosmos DB[9][10] от Microsoft.

Cosmos DB

Система управления базами данных (СУБД) – это программное обеспечение для создания баз данных, хранения и поиска в них необходимой информации. Существует много типов СУБД, нас интересуют два из них: реляционные (работает с систематизированными двумерными таблицами) и нереляционные (работает с иерархическими структурами данных, такими как json файлы). Именно поэтому мы используем базу данных Cosmos DB в которой при помощи простых запросов на sql-подобном языке можно получать сложные результаты. Они также хранятся в виде json файлов, но могут быть преобразованы в таблицы для удобной визуализации.

После загрузки данных в Cosmos DB, с ними можно работать в так называемых ноутбуках на sql-подобном языке и языке python.

Пример того что, можно сделать при помощи Cosmos DB:

```
SELECT p.title, r.source.text
FROM papers p JOIN r IN p.relations
WHERE r.relationType='DosageOfMedication' AND CONTAINS(r.target.text, 'hydroxychloroquine')
```

При помощи данного запроса мы получаем таблицу с названиями статей и приведенными в них дозировками медикамента гидроксихлорохина.

	title	text
0	In Vitro Antiviral Activity and Projection of ...	400 mg
1	In Vitro Antiviral Activity and Projection of ...	maintenance dose
2	In Vitro Antiviral Activity and Projection of ...	200 mg
3	An open-label prospective observational study ...	400mg/day
4	Clinical experience with therapeutic dose of L...	200mg
...	...	...
384	Prescribing practices of lopinavir/ritonavir, ...	increase
385	Prescribing practices of lopinavir/ritonavir, ...	32
386	Prescribing practices of lopinavir/ritonavir, ...	6
387	COVID-19 pandemic and biological therapy in rh...	dose
388	COVID-19 pandemic and biological therapy in rh...	400 mg

389 rows × 2 columns

Power BI

К сожалению, не каждый медицинский работник может свободно писать код на sql, поэтому для визуализации данных можно использовать сервис Power BI. При помощи sql запросов мы один раз загрузили данные из Cosmos DB, прописали их взаимосвязи и на

выходе получили систематизированную базу данных, с которой намного проще и понятнее работать. Пример таблицы дозировки медикаментов используя Power BI

Document.relationType	Сумма Count	Document.relationType	Document.source	Document.target	Count
Abbreviation	713475	DosageOfMedication	therapeutic	anticoagulation	68
DirectionOfBodyStructure	12675	DosageOfMedication	400 mg	Hydroxychloroquine	51
DirectionOfCondition	13437	DosageOfMedication	400 mg	HCQ	49
DirectionOfExamination	3391	DosageOfMedication	high-dose	corticosteroids	49
DirectionOfTreatment	4276	DosageOfMedication	200 mg	Hydroxychloroquine	46
DosageOfMedication	23760	DosageOfMedication	high-dose	steroids	44
FormOfMedication	6510	DosageOfMedication	low-dose	aspirin	41
FrequencyOfMedication	4903	DosageOfMedication	500 mg	azithromycin	34
FrequencyOfTreatment	6921	DosageOfMedication	1	Angiotensin	33
QualifierOfCondition	459129	DosageOfMedication	200 mg	HCQ	29
RelationOfExamination	144470	DosageOfMedication	8 mg/kg	tocilizumab	28
RouteOfMedication	12612	DosageOfMedication	high dose	corticosteroids	28
TimeOfCondition	145256	DosageOfMedication	100 mg	remdesivir	24
TimeOfEvent	27825	DosageOfMedication	100 mg	ritonavir	24
TimeOfExamination	101553	DosageOfMedication	200 mg	remdesivir	24
TimeOfMedication	14981	DosageOfMedication	low-dose	corticosteroids	24
TimeOfTreatment	59917	DosageOfMedication	1-7	Angiotensin	23
UnitOfCondition	41130	DosageOfMedication	high-dose	corticosteroid	23
UnitOfExamination	279532	DosageOfMedication	600 mg	Hydroxychloroquine	22
ValueOfCondition	58324	DosageOfMedication	high dose	steroids	22
ValueOfExamination	853503	DosageOfMedication	high-dose	methylprednisolone	22
Bcero	2987580	DosageOfMedication	therapeutic dose	anticoagulation	22
		DosageOfMedication	4 mg	baricitinib	21

Таблица всех диагнозов и количество их упоминаний

Document.category	Count	Document.text	Count	Document.umls_id
AdministrativeEvent	171937	COVID-19	304388	C5203670
Age	155693	SARS-CoV-2	66542	C5203670
BodyStructure	245787	disease	62883	C0012634
CareEnvironment	130770	infection	40604	C3714514
ConditionQualifier	486762	pandemic	39095	C1615608
Date	42153	COVID-19 pandemic	38565	C5203670
Diagnosis	2477847	Coronavirus disease 2019	26917	C5203670
Direction	34398	death	17375	C0011065
Dosage	35550	infections	15596	C3714514
ExaminationName	2226245	SARS-CoV-2 infection	14599	C5203670
FamilyRelation	80454	deaths	14471	C0011065
Frequency	21274	SARS	14041	C1175175
Gender	67145	cancer	13526	C0006826
GeneOrProtein	39782	diseases	12976	C0012634
HealthcareProfession	120570	pneumonia	12840	C0032285
MeasurementUnit	384921	virus	12221	C0042769
MeasurementValue	1030936	influenza	11166	C0021400
MedicationClass	242516	illness	8764	C0221423
MedicationForm	7697	diabetes	8744	C0011849
MedicationName	297463	stress	8713	C0038435
MedicationRoute	16988	COVID-19 infection	8254	C5203670
RelationalOperator	190191	anxiety	8051	C0003467
SymptomOrSign	1117712	ARDS	7706	C0035222
Time	336369	complications	7609	C0009566
TreatmentName	1314612	coronavirus	7595	C0206750
Variant	4427	infected	7370	C3714514
		acute respiratory syndrome coronavirus 2	7237	

При помощи таких таблиц уже можно легко получать доступ к конкретным сущностям и делать различные выводы.

Построение графиков изменения тактик лечения от времени

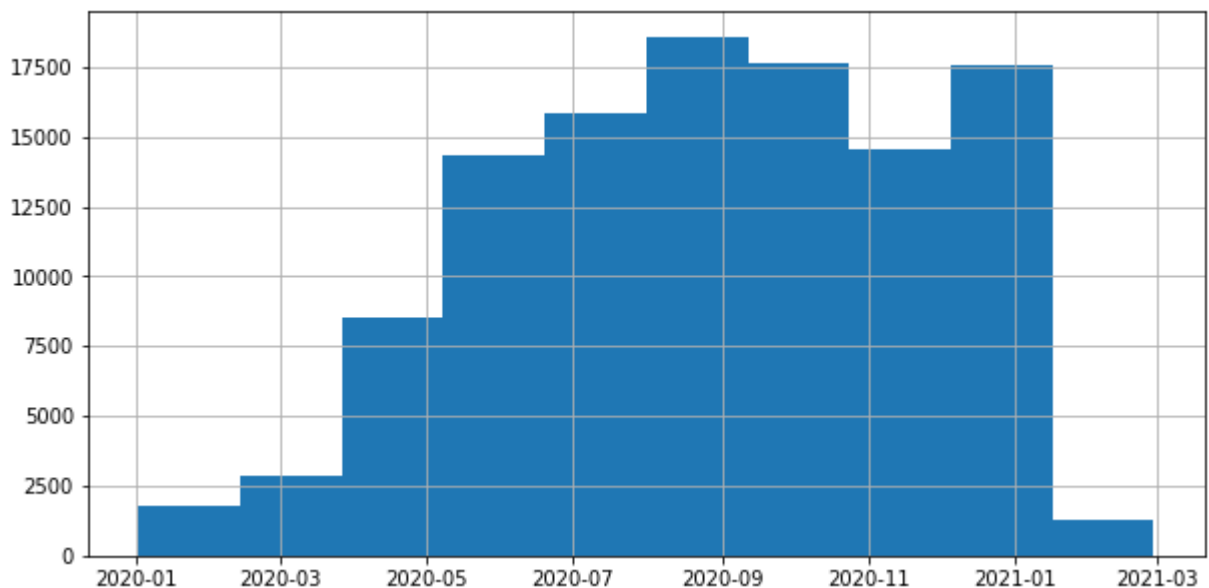
После изучения всех данных нам стало интересно, как изменялись тактики лечения и применяемые препараты с течением времени. При помощи Jupiter notebooks и ноутбуков, встроенных в Cosmos DB

Из всех сущностей при помощи sql-запроса мы выделили названия медикаментов и принадлежащие им параметры: название, отрицательный ли он имеет смысл, название статьи в которой оно упоминается, время публикации статьи и id в медицинской онтологии UMLS.

```
[11] 1 meds.head()
```

	text	isNegated	title	publish_time	umls_id
0	FIPV-3CL	False	X-Ray structure and inhibition of the feline i...	2015-11-01	NaN
1	FIPV-3CL	False	X-Ray structure and inhibition of the feline i...	2015-11-01	NaN
2	FIPV-3CL	False	X-Ray structure and inhibition of the feline i...	2015-11-01	NaN
3	coronaviral	False	X-Ray structure and inhibition of the feline i...	2015-11-01	C0206419
4	flew	False	Potential for the International Spread of Midd...	2013-07-17	C0441659

Далее мы построили таблицу зависимости количества статей от даты публикации. Можно заметить, что пик количества опубликованных статей приходится на сентябрь 2020 года. Это может быть обусловлено тем, что множество исследований требуют время для их проведения, следовательно, результаты многих исследований начатых в начале года будут опубликованы ближе к осени.



Нам показалось интересным рассмотреть динамику количества упоминаний конкретных лекарств, что позволяет рассмотреть изменение подходов лечения со временем. Первым шагом мы выделили самые упоминаемые лекарства. Для того чтобы предотвратить повторное упоминание медикамента под разными названиями мы использовали их привязку к UMLS id.

	title	isNegated	negativity	name
umls_id				
C0020336	4846	191	0.039414	hydroxychloroquine
C0008269	1870	38	0.020321	chloroquine
C1609165	1793	94	0.052426	Tocilizumab
C4726677	1625	24	0.014769	remdesivir
C0052796	1201	84	0.069942	azithromycin
...	...	...	...	...
C0282549	1	0	0.000000	HL-60 cells
C0006051	1	0	0.000000	Botulinum neurotoxin B
C0282545	1	0	0.000000	phacoemulsification
C0036774	1	0	0.000000	bovine serum albumin
C0067874	1	0	0.000000	1-butanethiol

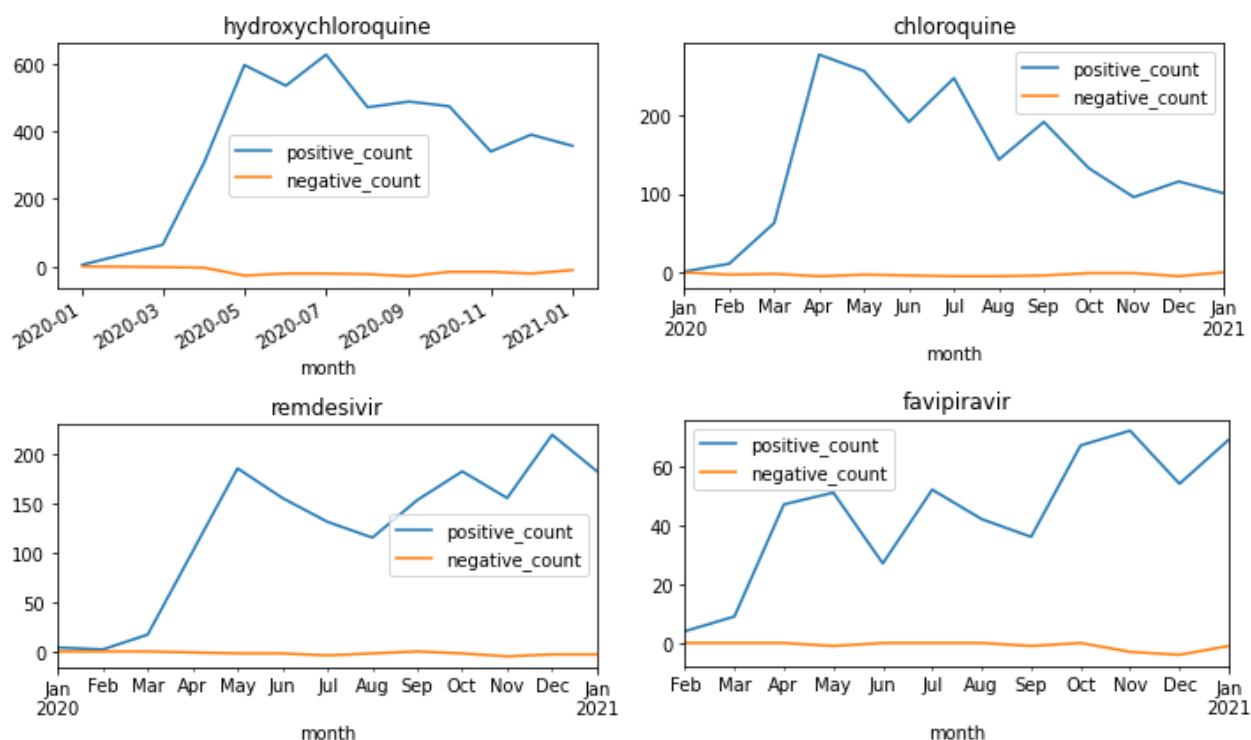
7879 rows × 4 columns

Далее для наиболее упоминаемых лекарств мы подсчитали их частоту упоминания по месяцам и построили графики.

		text	isNegated	
		count	positive_count	negative_count
month	name			
2020-01-01	Heparin	4	4	0
	azithromycin	3	3	0
	chloroquine	1	1	0
	hydroxychloroquine	5	5	0
	lipopolysaccharide	9	9	0
...	...	...	...	...
2021-01-01	methylprednisolone	40	36	-4
	ozone	118	118	0
	remdesivir	185	182	-3
	ritonavir	76	76	0
	vitamin D	79	77	-2

196 rows × 3 columns

Примеры графиков изменения количества упоминаний конкретных лекарств от даты публикации статьи



Как видно из первого графика, в начале года ученые активно изучали гидроксихлорохин, но позже частота упоминаний идет на спад, а количество упоминаний известного в России фавипиравира (последний график) со временем только растет.

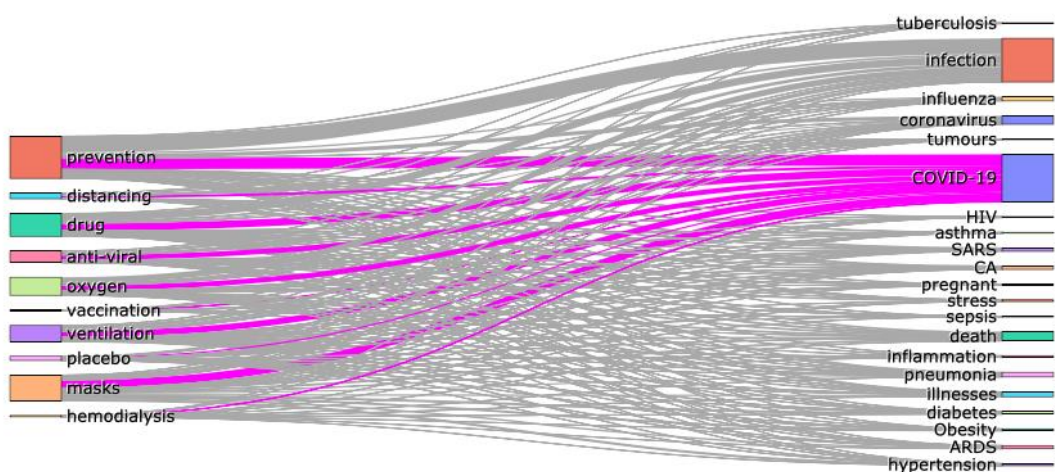
Графики взаимного упоминания терминов

Многие медицинские термины взаимосвязаны и вместе упоминаются в статьях. Так как мы можем найти взаимные упоминания, мы можем проследить зависимости одного или нескольких классов терминов, например, зависимость поставленного диагноза и медикамента для ее лечения. Для выявления таких зависимостей отлично подходят графики типа Sankey и хордовые диаграммы.

Для построения графиков мы использовали Jupiter notebooks, библиотеку Plotly для графиков Sankey[11] и специальную библиотеку Chord[12][13] для хордовых диаграмм.

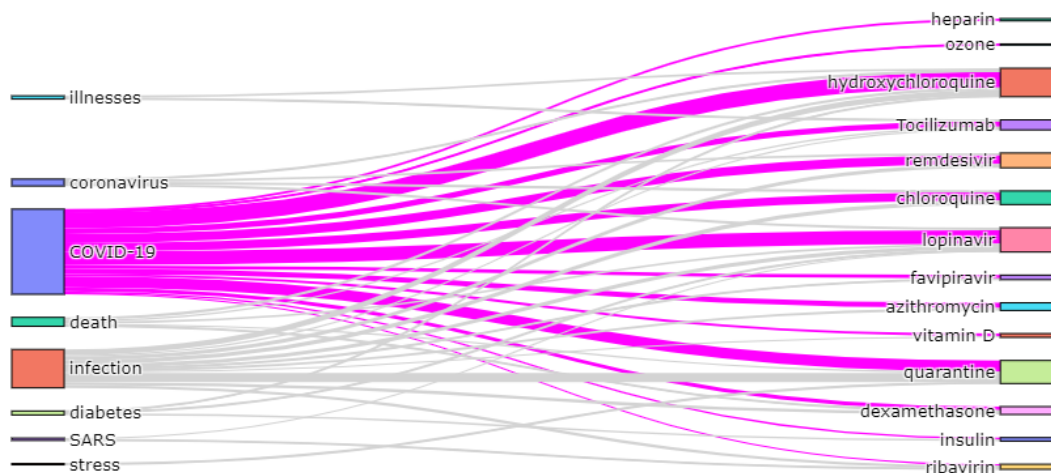
Функция графика Sankey принимает на вход массив названий узлов и 3 массива чисел: откуда идет линия, куда она ведет и значение. Толщина линии, соединяющей две сущности отображает значение. Цветом выделен COVID-19, так как он интересует нас больше всего. Для получения массивов мы построили корреляционные матрицы (см. приложение 3) по Json файлам. Поскольку на графике сложно отобразить много сущностей, мы вручную распределили схожие понятия и создали свою небольшую онтологию понятий. (см. приложение 1)

Взаимосвязь между способами лечения и диагнозами



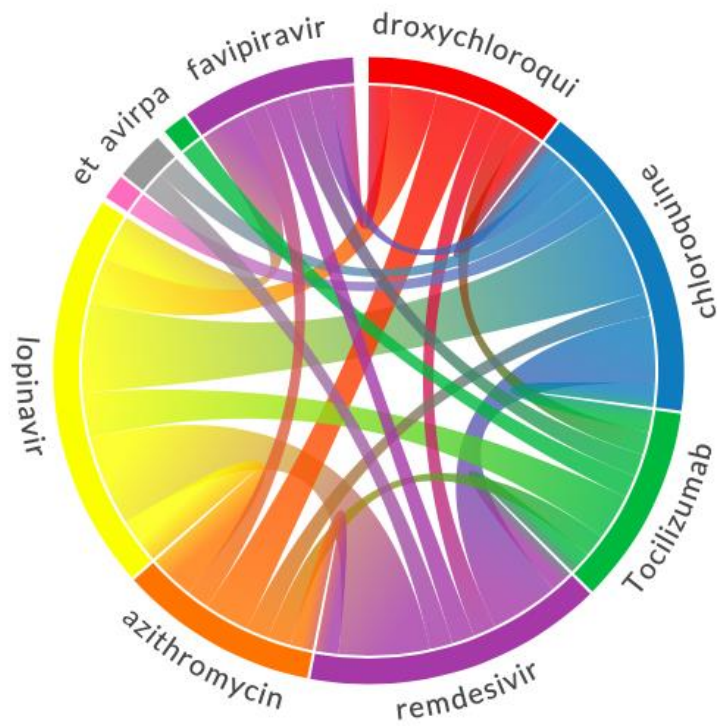
Можно заметить, что COVID-19 связан со всеми тактиками лечения, но в большей степени с предотвращением заражения, социальным дистанцированием и ношением масок

Взаимосвязь между диагнозами и медикаментами

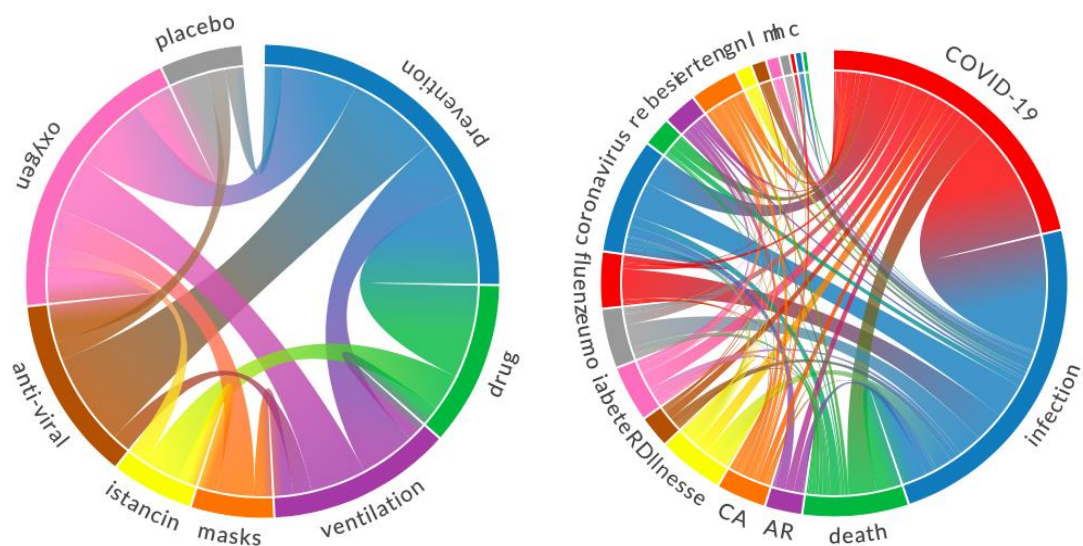


На данном графике больше всего выделена связь коронавирусной инфекции с гидроксихлорохином и лопинавиром, одними из самых упоминаемых лекарств во всех статьях, что также можно было заметить из таблиц Power BI. (см. приложение 2)

При помощи хордовых диаграмм мы построили графики взаимного упоминания терминов в пределах одного класса, например, для медикаментов получился следующий график:



Из этого графика можно увидеть взаимосвязь медикаментов, например, достаточно хорошо прослеживается пара гидроксихлорохин и азитромицин — препараты, которые используют совместно при лечении COVID-19.



Способы лечения и Диагнозы

Заключение

По итогам проделанной работы, мы предложили подход для извлечения полезной информации из больших объемов научных статей, который будет полезен исследователям как в области биомедицины, так и в других областях. Данный подход позволяет выделить из текста смысловые элементы и взаимосвязи между ними, а затем делать к ним семантические запросы и статистически их обрабатывать. Мы надеемся, что наша работа поможет ученым со всего мира ускорить свою работу.

Список литературы:

[1] How to: Use Text Analytics for health (preview) <https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/how-tos/text-analytics-for-health?tabs=ner>

[2] Wang L. L. et al. Cord-19: The covid-19 open research dataset //ArXiv. – 2020.

[3] Jacob Devlin, Ming-Wei Chang et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding – 2019

[4] Hakala K., Pyysalo S. Biomedical named entity recognition with multilingual BERT //Proceedings of The 5th Workshop on BioNLP Open Shared Tasks. – 2019. – С. 56-61.

[5] Peng Y., Chen Q., Lu Z. An empirical study of multi-task learning on BERT for biomedical text mining //arXiv preprint arXiv:2005.02799. – 2020.

[6] Lee, Jinhyuk et al. “BioBERT: a Pre-Trained Biomedical Language Representation Model for Biomedical Text Mining.” Ed. Jonathan Wren. Bioinformatics (2019): n. pag. Crossref. Web.

[7] BiomedNLP-PubMedBERT <https://huggingface.co/microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract>

[8] PubMed <https://pubmed.ncbi.nlm.nih.gov/>

[9] Paz J. R. G. Introduction to azure cosmos db //Microsoft Azure Cosmos DB Revealed. – Apress, Berkeley, CA, 2018. – С. 1-23.

[10] Azure Cosmos DB. Fast NoSQL database with open APIs for any scale <https://azure.microsoft.com/en-us/services/cosmos-db/>

[11] Sankey Diagram in Python <https://plotly.com/python/sankey-diagram/>

[12] Create beautiful and interactive Chord Diagrams using Python <https://towardsdatascience.com/create-beautiful-and-interactive-chord-diagrams-using-python-cb5ecb092a7c?gi=2cf8de15f7bd>

[13] Chord PyPI <https://pypi.org/project/chord/>

[14] Steven Bird, Ewan Klein, and Edward Loper “Natural Language Processing with Python” book O'Reilly Media, 2009

[15] CORD-19 Research Dataset: Analysis & Visualization <https://www.kaggle.com/finalepoch/cord-19-research-dataset-analysis-visualization>

Приложения:

1. Описание нашей микроонтологии

```
treatment_ontology = {  
  'C0042196': ('vaccination',1),  
  'C0199176': ('prevention',2),  
  'C0042210': ('vaccines',1),  
  'C0013227': ('drug',3),  
  'C0199470': ('ventilation',4),  
  'C0024861': ('masks',5),  
  'C5392135': ('distancing',6),  
  'C0021925': ('intubation',4),  
  'C0015357': ('ECMO',4),  
  'C0020971': ('immunization',1), ...}
```

2. Таблица медикаментов, отсортированная по количеству упоминаний

Document.category	Count	Document.text	Count	Document.umls_id
AdministrativeEvent	171937	hydroxychloroquine	5889	C0020336
Age	155693	HCQ	3667	C0020336
BodyStructure	245787	remdesivir	3188	C4726677
CareEnvironment	130770	chloroquine	2989	C0008269
ConditionQualifier	486762	Tocilizumab	2726	C1609165
Date	42153	lopinavir	2004	C0674432
Diagnosis	2477847	azithromycin	1980	C0052796
Direction	34398	quarantine	1890	C0034386
Dosage	35550	ritonavir	1871	C0292818
ExaminationName	2226245	cytokine storm	1499	
FamilyRelation	80454	insulin	1202	C0021641
Frequency	21274	ribavirin	1147	C0035525
Gender	67145	dexamethasone	1118	C0011777
GeneOrProtein	39782	CQ	1081	C0008269
HealthcareProfession	120570	vitamin D	1081	C0042866
MeasurementUnit	384921	favipiravir	1063	C1138226
MeasurementValue	1030936	heparin	979	C0019134
MedicationClass	242516	PEDV	946	C0318855
MedicationForm	7697	Cochrane	934	
MedicationName	297463	propofol	879	C0033487
MedicationRoute	16988	ozone	767	C0030106
RelationalOperator	190191	Aspirin	753	C0004057
SymptomOrSign	1117712	cytokine	750	C0079189
Time	336369	methylprednisolone	742	C0025815
TreatmentName	1314612	Oseltamivir	742	C0874161
Variant	4427	COVID-19	685	C5203676
		Mpro	679	

3. Код для построения корреляционной матрицы

```
def get_entities(x):
    s = set()
    for i in x['entities']:
        for j in i.get('links',[]):
            if j['dataSource'] == 'UMLS':
                s |= {j['id']}
    return s

import numpy as np
def get_matrix(cat1, cat2, l):
    d1 = {i:j[1] for i,j in cat1.items()}
    d2 = {i:j[1] for i,j in cat2.items()}
    s1 = set(cat1.keys())
    s2 = set(cat2.keys())
    a = np.zeros((len(cat1),len(cat2)))
    for i in l:
        ent = get_entities(i)
        for j in ent & s1:
            for k in ent & s2 :
                a[d1[j],d2[k]] += 1
    return a
```