СРАВНЕНИЕ МЕТОДОВ ПОИСКА ИМЕНОВАННЫХ СУЩНОСТЕЙ НА ПРИМЕРЕ РЕЙТИНГА КОМПАНИЙ ПО КОЛИЧЕСТВУ ПУБЛИКАЦИЙ НА ТЕМУ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Даргель Юлия
ВНИУ ВШЭ ДПО

Компьютерная лингвистика

СОДЕРЖАНИЕ

АКТУАЛЬНОСТЬ ПРОЕКТА

Оценка влияния генеративного ИИ по некоторым отраслям экономики РФ

	Наименование отраслей	Оценка структуры ВВП РФ, % в 2024 г.	Оценка ВВП в текущих ценах, млрд руб., 2024 г.	Оценка влияния генеративного ИИ на ВВП РФ по отраслям, %		Оценка влияния генеративного ИИ на ВВП РФ по отраслям, млрд руб	
				от	до	от	до
1.	Обрабатывающие производства	13,4	23 773,8	0,5	1,1	118,9	261,5
2.	Добыча полезных ископаемых	12,2	21 644,8	0,3	1,0	64,9	216,4
3.	Электроэнергетика и водоснабжение	2,6	4612,8	0,2	0,6	9,2	27,7
4.	Торговля оптовая и розничная	11,9	21 112,6	1,0	2,2	211,1	464,5
5.	Транспортировка и хранение	5,6	9935,3	0,6	1,0	59,6	99,4
6.	Сельское хозяйство	3,6	6387,0	0,2	0,5	9,6	31,9
7.	Финансовая и страховая сферы	4,2	7451,5	1,4	2,8	104,3	208,6
8.	Наука и образование	6,8	12 064,3	0,9	1,8	108,6	217,2
9.	Медицина и здравоохранение	3,4	6 032,2	1,2	2,6	72,4	156,8
10.	Строительство и недвижимость	14,0	24 838,3	0,3	0,9	74,5	223,5
11.	Телеком и связь, медиа (вкл. рекламу)	2,4	4258,0	1,8	3,1	76,6	132,0
12.	Госсектор (военная безопасность, соцобеспечение)	7,0	12 419,2	0,5	1,0	62,1	124,2
13.	Прочее	12,9	22 886,7	0,0	0,1	6,9	11,4
	ИТОГО	100,0	177 416,5			978,7	2 175,2

Источник: Росатом на Хабре

- Искусственный интеллект — «горячая» и хайповая технология
- Многие о ней говорят
- Мало кто реально делает
- А если делает, то не всегда рассказывает публично

Важно понять, кто рассказывает в публичном пространстве о реальных проектах, а не просто дает пустые комментарии

ЦЕЛИ

- Разработать алгоритм, который позволит выяснить, какие российские компании больше других говорят о технологии ИИ в информационном пространстве
- Сравнить результат собственного алгоритма с результатом применения готовой библиотеки для поиска именованных сущностей

ЗАДАЧИ ПРОЕКТА

- Разработать методику, на основе которой будет выполнятся алгоритм составления рейтинга
- Собрать корпус текстов
- Проверить гипотезы, сформированные при разработке методики
- Применить итоговый алгоритм для составления рейтинга
- Визуализировать результаты
- Сделать выводы
- Изучить применение готовых библиотек для поиска именованных сущностей
- Применить один из инструментов на собранном корпусе
- Провести сравнение результатов

МЕТОДИКА
И АЛГОРИТМ ДЕЙСТВИЙ

СБОР КОРПУСА

Для анализа были выбраны статьи на портале Comnews.ru за 2022-2024 гг.

- Comnews — экспертное отраслевое издание, посвященное ИТ и телекому. Редакция проводит качественный отбор материалов, не ставит кликбейтные новости и не использует «временное размещение» пресс-релизов (в отличие от Cnews).
- Публикация на портале доступна для компаний разного размера (в отличие от TIER-1 СМИ, как Коммерсант, Ведомости, Forbes)
- За три года на портале появилось около 1500 материалов, посвященных ИИ

```
[ ]  pages = ["https://www.comnews.ru/search?text=%D0%B8%D0%B8"]
     for i in range(1, 80): #на момент сбора дата-сета публикации с 2024 по 2022 гг располагались до 79 страницы поисковой выдачи
       pages.append('https://www.comnews.ru/search?text=%D0%B8%D0%B8&page=' + str(i))

     pages
```

```
news = [] #список ссылок на публикации, генерится около 15 мин


for i in pages:
  page = requests.get(i)
  soup = BeautifulSoup(page.text)

  for hr in soup.find_all('a', {"class" : "srch-row"}):
    news.append('https://www.comnews.ru' + hr.get('href'))

print(news[:2]) #пример того, что нагенерилось
print(len(news)) #исходное количество страниц для парсинга
```

```
['https://www.comnews.ru/content/237772/2025-02-14/2025-w07/1007/tureckiy-rynok-primet-rossiyskie-it-resheniya-lish
1600
```

```
[ ]  %%time

     data = [] # сбор данных занимает около 1 ч 40 мин

     for article in news:
       try:
         page = requests.get(article)
         soup = BeautifulSoup(page.text, features="html.parser")
         date =  soup.find('div', {'class' : 'field field-text field-name-date'}).text #дата
         if '2024' in date or '2023' in date or '2022' in date: #оставляем только наши три года
           title = soup.find('h1').text # заголовок
           text = soup.find('div', {'class' : 'field field-text full-html field-name-body'}).text
           data.append((date, title, text, article))
         time.sleep(1)
       except:
         print('Со страницей ' + article + ' не работает')
     data
```

```
[ ]  df = pd.DataFrame(data)
     df.columns = ['date', 'title', 'text', 'link']
     df
```

	date	title	text	link
0	25.12.2024	На Открытой конференции ИСП РАН 2024 обсудили ...	1500 участников собрала Открытая конференция И...	https://www.comnews.ru/content/237018/1018/
1	24.12.2024	BIA Technologies: автоматизация, цифровые двой...	Развитие цифровых решений для логистики, торго...	https://www.comnews.ru/content/237017/1018/
2	24.12.2024	Киберэксперт Дмитрий Овчинников назвал топ-3 с...	В новогодние праздники мошенники традиционно а...	https://www.comnews.ru/content/237016/1018/
3	24.12.2024	Письмо Деду Морозу от редакции ComNews	Дорогой Дедушка Мороз! Пишет тебе редакция Com...	https://www.comnews.ru/content/236941/2024-12-...

АЛГОРИТМ ПОИСКА КОМПАНИЙ

Был выбран следующий алгоритм:

- Проводим предобработку текста без приведения к нижнему регистру (удаление пунктуации и стоп-слов, токенизация)
- Находим слова, которые соответствуют условиям:
  - начинаются с заглавной буквы
  - входят в текст публикации не менее 3 раз (обоснование — в разделе «Гипотезы»)
- Проводим грамматический анализ выбранных кириллических слов, удаляем лишние по критериям:
  - одушевленное существительное
  - служебная часть речи (предлог, местоимение и др) или прилагательное
- Считаем частотность.
- Анализируем топ-50, при необходимости «дочищаем» (удаляем иностранные компании, названия министерств и т.д.)
- Визуализируем результат

```
[ ]  # Выполняем предобработку и находим слова, начинающиеся с заглавной буквы

     def up_words(text):
       names = []
       uppers = []
       stop_words = stopwords.words('russian')
       morph = MorphAnalyzer()
       text_no_punkt = re.sub('[^\w\d -]', '', text) #удаляем пунктуацию
       text_list_nltk = word_tokenize(text_no_punkt) #токенизируем
       text_clean = [word for word in text_list_nltk if word not in stop_words and word[0].isalpha()] #удаляем стоп-слова и "мусор"
       for word in text_clean:
         if word[0].isupper(): #находим слова с первой загловной буквой
           lemm = morph.parse(word)[0].normal_form #лемматизируем
           names.append(lemm)
       word_frequencies = FreqDist(names) #считаем частоту
       for token, frequency in word_frequencies.items():
         if frequency >= 3: #если частота больше 3, предполагаем, что слово может нам подойти
           uppers.append(token)
       return uppers
```

```
[ ]  #собираем список одушевленных сущ и географичексих названий
     names = []
     #при анализе первичного списка выяснилось, что нужно внести исключения:
     exceptions = ['сбер', 'сберсити', 'сберкорус', 'авить', 'нейролаб', 'касперский', 'сегежа', 'рольф', 'спортмастер', 'эвотор', 'протей' , 'ростсельмаш
     for word in text_mystem:
       if word['text'] not in exceptions:
         if word.get('analysis'):
           if ',од=' in word['analysis'][0]['gr'] or 'reo' in word['analysis'][0]['gr']:
             names.append(word['text'])
     print(names)
```

```
['рф', 'дмитрий', 'россия', 'ия', 'евгений', 'дмитрий', 'овчинник', 'мошенник', 'дедушка', 'ия', 'дед', 'ия', 'ия', 'екатерина', 'коныгин', 'максим',
```

```
[ ] df['uppers_new'] = df['uppers'].apply(uppers_clean) #создаем новый столбец с очищенным списком брендов
    df
```

	date	title	text	link	uppers	uppers_new
0	25.12.2024	На Открытой конференции ИСП РАН 2024 обсудили ...	1500 участников собрала Открытая конференция И...	https://www.comnews.ru/content/237018/1018/	[исп, ран, академия, рф, в, дмитрий, россия, и...	[исп, ран, академия]
1	24.12.2024	BIA Technologies: автоматизация, цифровые двой...	Развитие цифровых решений для логистики, торго...	https://www.comnews.ru/content/237017/1018/	[кроме, bia, technologies]	[bia]
2	24.12.2024	Киберэксперт Дмитрий Овчинников назвал топ-3 с...	В новогодние праздники мошенники традиционно а...	https://www.comnews.ru/content/237016/1018/	[дмитрий, овчинник, мошенник]	[]
3	24.12.2024	Письмо Деду Морозу от редакции ComNews	Дорогой Дедушка Мороз! Пишет тебе редакция Com...	https://www.comnews.ru/content/236941/2024-12-...	[дедушка, мороз, в, ия, дед, но]	[мороз]

```
[ ] #Формируем рейтинг

    AI_rating = FreqDist(brands)
    AI_rating.most_common(57)

    # 5 и более публикаций у 57 компаний.
    #Нижние строчки (1-4 публ) не показательны, поэтому не вычищались
```

ГОТОВЫЙ NER

- Находим названия организаций, которые встречаются в 1 публикации не менее 3 раз (обоснование — в разделе «Гипотезы»)
- Считаем частотность.
- Анализируем топ-50, при необходимости «дочищаем» (удаляем иностранные компании, названия министерств и т.д.)
- Визуализируем результат

```
[ ]  #Функция для поиска именованных сущностей в столбце датасета, которые повторяются не менее 3 раз + лемматизация
     def co_names_sp(text):
       co_names = []
       org = []
       doc = load_model(text)
       stop_words = stopwords.words('russian')
       morph = MorphAnalyzer()
       for entity in doc.ents:
         if entity.label_ == 'ORG': #находим названия организаций
           try:
             org_nltk = word_tokenize(entity.text) #токенизируем кириллические названия
             org_clean = [word for word in org_nltk if word not in stop_words and word[0].isalpha()] #удаляем стоп-слова и "мусор"
             for word in org_clean:
               lemm = morph.parse(word)[0].normal_form #лемматизируем кириллические названия
               co_names.append(lemm)
           except:
             co_names.append(entity.text) #подхватываем англоязычные названия
       org_frequencies = FreqDist(co_names) #считаем частоту в статье
       for token, frequency in org_frequencies.items():
         if frequency >= 3: #если частота больше 3, предполагаем, что слово может нам подойти
           org.append(token)
       return org
```

```
[ ] df['org_names'] = df['text'].apply(co_names_sp) #создаем новый столбец с названиями. Занимает около 12 мин
    df
```

	date	title	text	link	org_names
0	25.12.2024	На Открытой конференции ИСП РАН 2024 обсудили ...	1500 участников собрала Открытая конференция И...	https://www.comnews.ru/content/237018/1018/	[исп, ран, центр, криптография, академия, инст...
1	24.12.2024	BIA Technologies: автоматизация, цифровые двой...	Развитие цифровых решений для логистики, торго...	https://www.comnews.ru/content/237017/1018/	[bia, technologies]
2	24.12.2024	Киберэксперт Дмитрий Овчинников назвал топ-3 с...	В новогодние праздники мошенники традиционно а...	https://www.comnews.ru/content/237016/1018/	[]
3	24.12.2024	Письмо Деду Морозу от редакции ComNews	Дорогой Дедушка Мороз! Пишет тебе редакция Com...	https://www.comnews.ru/content/236941/2024-12-...	[]
4	23.12.2024	Экзоскелеты с искусственным интеллектом разраб...	Компании ООО "Экзо Солюшенс" и "Социальный код...	https://www.comnews.ru/digital-economy/content...	[]

```
[ ] #Как и в авторском варианте, в первоначальном списке много мусора. Поэтому чистим вручную
    exclude = ['ооо', 'центр', 'цифровой', 'пао', 'ано', 'университет', 'comnews', 'гк', 'ао', 'россия', 'engines', 'group', 'лаборатория'
    print(len(exclude)) #исключений на 20% меньше, чем в авторском варианте
```

```
88
```

```
#собираем список компаний, которым посвящены публикации
brands = []
for group in df['org_names']:
  for word in group:
    if word not in exclude:
      brands.append(word)
print(brands)
```

```
['исп', 'ран', 'криптография', 'академия', 'исследование', 'системный', 'bia', 'магнит', 'remez', 'м.видео-эльдорадо', 'первый', 'битый
```

ИНСТРУМЕНТЫ

Сбор корпуса методом парсинга с помощью:

- BeautifulSoup
- Regex
- Pandas
- time

Реализация алгоритма по поиску названий компаний:

- Regex
- nltk
- pymorphy3
- Mystem
- Wordcloud

Сравнение с готовым NER:

- SpaCy

ПРОВЕРКА ГИПОТЕЗ

ЧТО ВЫЯСНИЛОСЬ

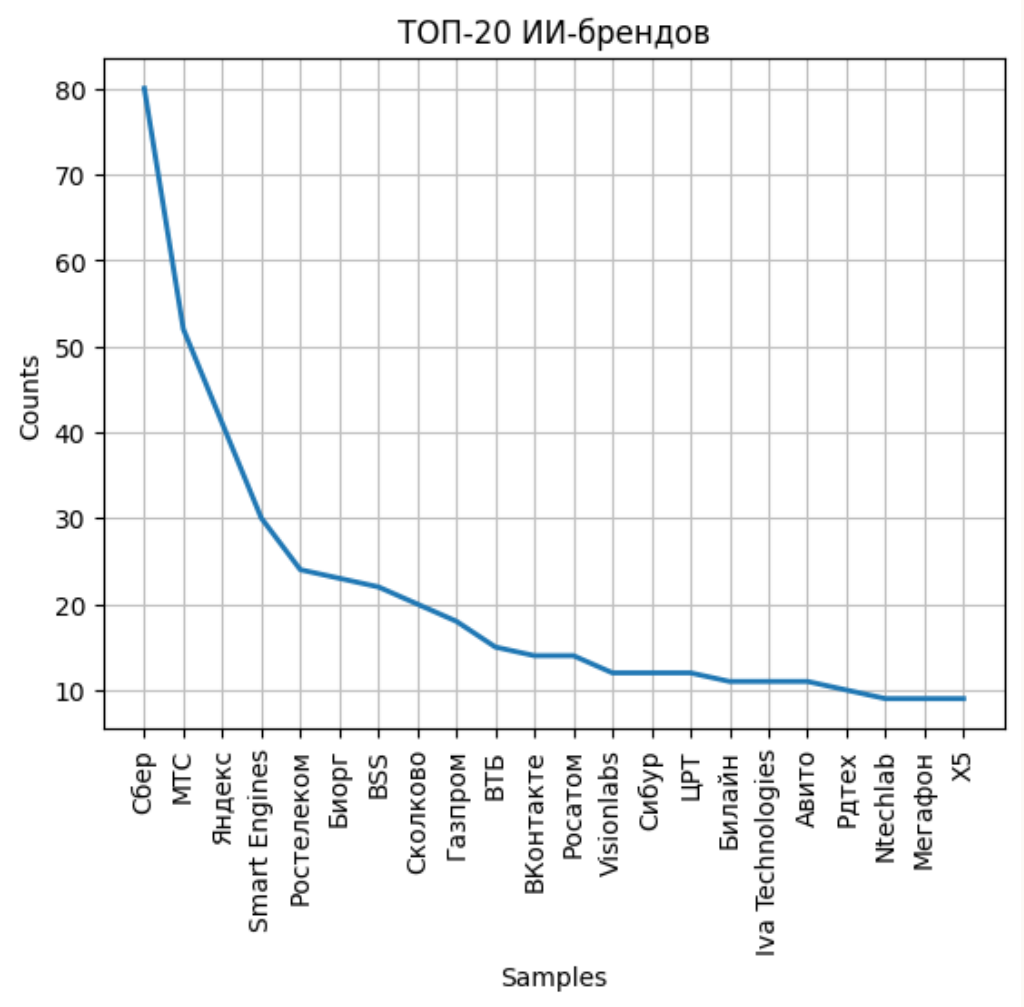
- ✓ Если слово начинается с заглавной буквы и входит в текст 3 раза и более, вероятно, это название компании, которой посвящена публикация.
  Как правило, ФИО спикеров упоминают 1-2 раза. Если упоминание только одно, скорее всего, компанию привели в качестве примера
- ✓ Названия компаний могут писаться по-разному, поэтому ручной очистки итогового рейтинга не избежать

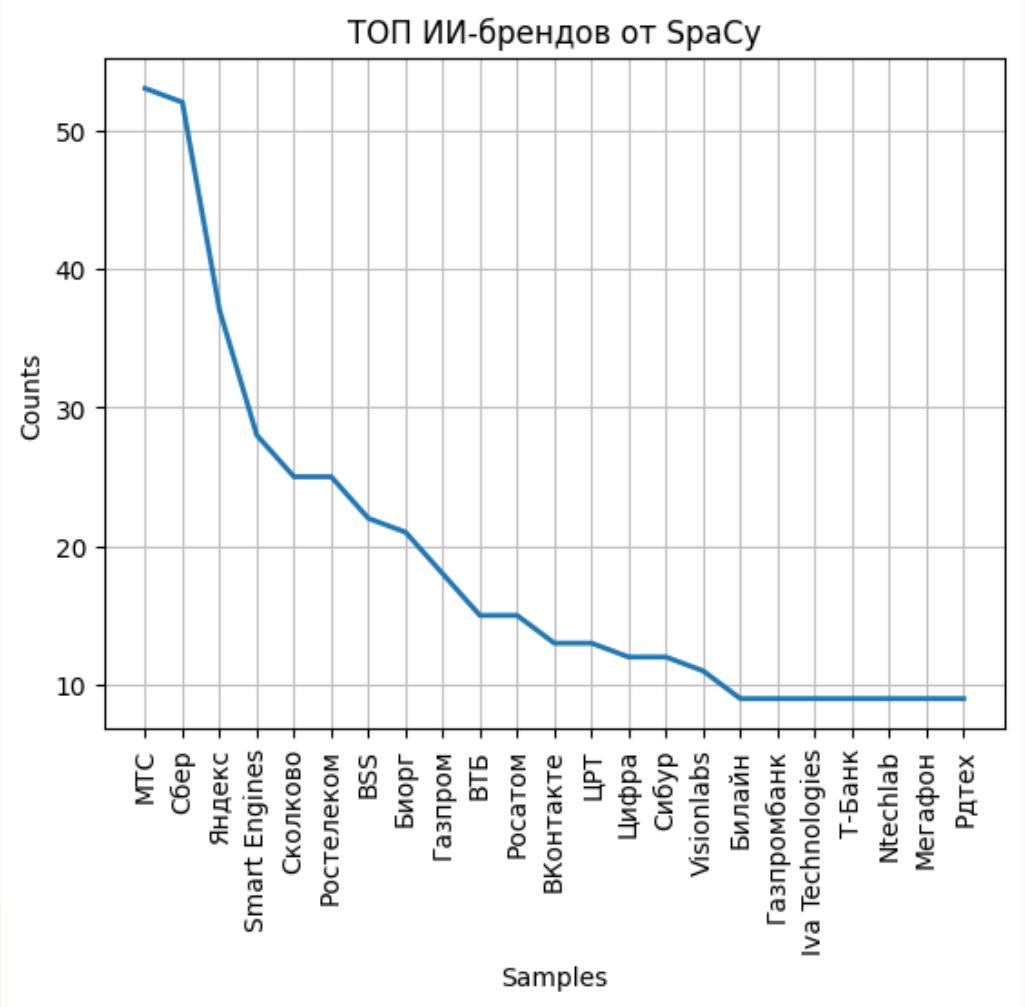
- ☒ С помощью регулярных выражений можно найти кириллически названия из нескольких слов в кавычках.
  По факту, выделяются цитаты
- ☒ Можно использовать поиск «регулярками» слов с заглавной первой буквой, стоящих не в начале строки
  По факту условия про 3 вхождения отсекает большинство слов начала предложения. Союзы и вводные слова удаляются на этапе грамматического разбора

РЕЗУЛЬТАТЫ

АВТОРСКИЙ ПОИСК VS SPACY

РАСПРЕДЕЛЕНИЕ ЧИСЛА ПУБЛИКАЦИЙ





СРАВНЕНИЕ ПОЗИЦИЙ

- 4 лидера сохраняются
- Сбер: разница в количестве публикаций в 2 раза (!)
- Расхождение для абсолютного большинства компаний минимальное – 1-2 публикации
- В авторский рейтинг не попал Банк Цифра
- Авито: в топ от Spacy всего 4 публикации, а в авторском – 11

Место	Авторский подсчет	Место	Spacy
1	('Сбер', 80),	1	('МТС', 53),
2	('МТС', 52),	2	('Сбер', 52),
3	('Яндекс', 41),	3	('Яндекс', 37),
4	('Smart Engines', 30),	4	('Smart Engines', 28),
5	('Ростелеком', 24),	5-6	('Сколково', 25),
6	('Биорг', 23),	5-6	('Ростелеком', 25),
7	('BSS', 22),	7	('BSS', 22),
8	('Сколково', 20),	8	('Биорг', 21),
9	('Газпром', 18),	9	('Газпром', 18),
10	('ВТБ', 15),	10-11	('ВТБ', 15),
11-12	('ВКонтакте', 14),	10-11	('Росатом', 15),
11-12	('Росатом', 14),	12-13	('ВКонтакте', 13),
13-15	('Visionlabs', 12),	12-13	('ЦРТ', 13),
13-15	('Сибур', 12),	14-15	('Цифра', 12),
13-15	('ЦРТ', 12),	14-15	('Сибур', 12),
16-18	('Билайн', 11),	16	('Visionlabs', 11),
16-18	('Iva Technologies', 11),	17-23	('Билайн', 9),
16-18	('Авито', 11),	17-23	('Газпромбанк', 9),
19	('Рдтех', 10),	17-23	('Iva Technologies', 9),
20-22	('Ntechlab', 9),	17-23	('Т-Банк', 9),
20-22	('Мегафон', 9),	17-23	('Ntechlab', 9),
20-22	('X5', 9),	17-23	('Мегафон', 9),
23-30	('РАН', 8),	17-23	('Рдтех', 9),

БОНУС

Что Mystem думает о компаниях:

- многие названия – одуш. сущ.
- гринат, уралхий, ростёха – интересный им. п.
- авить – гл. (!)

А еще график распределения публикаций по компаниям похож на закон Ципфа ☺

ВЫВОДЫ

Результат применения собственного алгоритма отражает реальное положением на рынке

- В ТОП-50 рейтинга попали компании из разных сфер:
  - Финансы (Сбер, ВТБ и др.)
  - Телеком (МТС, Ростелеком, билайн, Мегафон)
  - ИТ (BSS, Positive Technologies и др.)
  - Промышленность (Росатом, Норникель и др.)
- Результаты собственного алгоритма и NER от Spacy разнятся незначительно. Это говорит о том, что для составления общего представления о происходящем собственный алгоритм дает вполне объективные результаты.
- Направления дальнейшего совершенствования: расширить корпус публикаций за счёт других СМИ, объединять статьи по сюжетам, а также использовать заранее составленные списки компаний для более точной идентификации в публикациях

СПАСИБО!

Проект на GitHub:
https://github.com/Dargel/AI_Rating_NER_comparison

Код в Colab:
Часть 1 – Парсинг
Часть 2 – Авторский рейтинг
Часть 3 – Рейтинг со SpaCy