ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему:

# НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ КОМПЛЕКС ПРЕДОБРАБОТКИ ТЕКСТОВ И ПОСТОБРАБОТКИ РЕЗУЛЬТАТОВ В СИСТЕМЕ SKETCHENGINE

# Выпускная квалификационная работа по направлению

# 45.03.02 «Лингвистика»,

Исполнитель:

Обучающийся 4 курса

Образовательной программы

«Прикладная и экспериментальная

лингвистика»

очной формы обучения

Огай Андрей Александрович

Научный руководитель:

к.ф.н., доц. Захаров В.П.

Рецензент:

      к.ф.н. доц. Хохлова М.В.

# Санкт-Петербург

2018

Оглавление

[Введение 3](#_Toc515964291)

[Глава 1: Путь к веб-корпусу 5](#_Toc515964292)

[1. Этапы становления и развития корпусной лингвистики 5](#_Toc515964293)

[§3. Развитие программно-лингвистического инструментария 10](#_Toc515964294)

[Глава 2: Работа системы SketchEngine 14](#_Toc515964295)

[2.1. Предобработка 14](#_Toc515964296)

[2.1.1. Построение корпуса в SketchEngne 14](#_Toc515964297)

[2.1.2. Проблемы с лемматизацией 16](#_Toc515964298)

[2.2. Постобработка и визуализация 18](#_Toc515964299)

[2.2.1. Методы работы с корпусом. 18](#_Toc515964300)

[2.2.2. Меры и формулы. 20](#_Toc515964301)

[2.2.3. Решения проблем 22](#_Toc515964302)

[Глава 3: Рабочий процесс 23](#_Toc515964303)

[§3.1. Предобработка 23](#_Toc515964304)

[§3.2. Постобработка 25](#_Toc515964305)

[3.2.1. Извлечение терминов 25](#_Toc515964306)

[§3.2.2. Создание мер 28](#_Toc515964307)

[§3.2.3. Расширенные тезаурусы 30](#_Toc515964308)

[§3.3. Визуализация 32](#_Toc515964309)

[§3.3.1. Принципы визуализации 32](#_Toc515964310)

[§3.3.2. Визуализация расширенного тезауруса 33](#_Toc515964311)

[§3.3.3. Другие визуализации 36](#_Toc515964312)

[Заключение 38](#_Toc515964313)

[Литература 39](#_Toc515964314)

# Аннотации

Система SketchEngine (сайт sketchengine.co.uk), созданная британским лингвистом Адамом Килгариффом, изучает грамматическое и коллокационное поведение слов на основе данных веб-корпусов, показывая с помощью статистических таблиц распространенность в языке тех или иных связей между разными словами. Данная работа посвящена способам работы с этой системой: рассмотрены и показаны различные разработанные автором инструменты, позволяющие автоматически измерять и сравнивать полученную с помощью SketchEngine статистическую информацию, а также качественно визуализировать ее.

|  |
| --- |
| Ключевые слова: sketchengine,визуализация,корпусная лингвистика,статистика,коллокации |
|  |

The SketchEngine system (sketchengine.co.uk) created by British linguist Adam Kilgariff studies the grammatical and collocational behaviour of words based on the data collected from web corpora. Through statistic tables, it shows how frequent certain connections between words are in a language. This work is dedicated to the means of work with this system; multiple instruments developed by the author are shown, allowing to automatically compare and measure the statistical information gathered via SketchEngine, and to visualise it.

Keywords: sketchengine,visualisation,corpus linguistics,statistics,collocation

# Введение

Сайт SketchEngine, созданный британским исследователем Адамом Килгариффом, является одним из самых широко используемых инструментов корпусной лингвистики. Это система с мощным функционалом, позволяющая исследовать языковые корпуса с разных точек зрения (от грамматических и синтаксических отношений до семантических связей), находящая применение как и среди людей, далеких от мира языкознания (в обучении языку), так и среди профессиональных лексикографов и лингвистов. Одним из первых SketchEngine открыл возможности Интернета в корпусной лингвистике, дав возможность не только изучать уже готовые корпуса, но и создавать их самим на основе интернет-данных.

Именно из-за важности и широте использования SketchEngine так актуальны проекты по созданию вспомогательных инструментов для этой системы, позволявших бы более эффективно работать с корпусными данными, как, например, SKELL. Сама система, созданная Килгариффом, была создана для того чтобы пользователю было легче воспринимать информацию, полученную из корпусов; вполне логично, что другие люди будут развивать его идею и находить новые способы ее обработки и визуализации. Для SketchEngine существует достаточно развитый API, позволяющий пользователю свободно работать с корпусными данными и использовать их в собственных программах, что дает нам большие возможности по созданию своего инструментария.

Целью этой работы является создание такого инструментария - научно-исследовательского комплекса, который позволит нам уточнять полученную из SketchEngine информацию, показывать ее с других сторон, на которые ранее не обращалось внимания, а также качественно её визуализировать.

Чтобы достигнуть этой цели, необходимо будет выполнить три задачи. Во-первых, в пределах этой работы исследован процесс построения корпусов в системе SketchEngine, рассмотрены его сильные и слабые стороны. Известно, что даже сбалансированный по временному промежутку, источникам, языкам корпус может быть неполным и содержать большое количество “шума” (Gabrielatos, 2007) - что же говорить о пользовательских корпусах, строящихся *ad hoc*? На основе этих данных делаются выводы о возможных методах предобработки языковых данных перед их добавлением в корпус.

Второй по счету задачей является непосредственно работа со корпусными данными, а в особенности с мерами, которые они используют. За свою историю SketchEngine использовала самые разные способы выявления коллокации: разные методы, являющиеся частью системы, задействуют разные меры, каждая из которых дает свои результаты (Mi, T-Score, Log-likelihood, etc.) и которые в дальнейшем приходится соотносить вручную; лингвистическая оценка получаемых данных (Хохлова, 2010) зачастую занимает много времени и сил. В этой работе рассмотрены разработанные автором инструменты, которые дают возможность сделать методы SketchEngine более точными и расширять их возможности.

Наконец, существует не так много хороших средств по визуализации итоговых данных, что не позволяет исследователю получить наглядные результаты своей работы, по которым сразу возможно было бы делать определенные выводы. Визуализировав коллокационные данные самостоятельно, мы сможем сделать нашу итоговую работу гораздо более наглядной и понятной. Исследователь сможет увидеть информацию в организованном и маркированном виде, сразу заметить общие тренды и исключения из них и таким образом обнаружить новые вещи в уже имеющихся данных.

Теоретическая значимость работы определяется подробным описанием слабостей системы SketchEngine, ранее не затрагиваемых в научной работе. Практическая значимость данного исследования заключается в инструментарии, отчасти компенсирующем эти слабости и делающем работу над SketchEngine более эффективной.

Данная работа состоит из введения, трёх глав, заключения, списка литературы и приложения. В первой главе описываются теоретические вопросы, связанные с историей веб-корпусов. Вторая глава посвящена методологии конкретно системы SketchEngine, ее основным возможностям, сильным и слабым сторонам и тому, где ее возможно улучшить с помощью нашего инструментария. В третьей главе описан процесс работы над инструментарием и его возможности; представлена оценка полученных результатов.

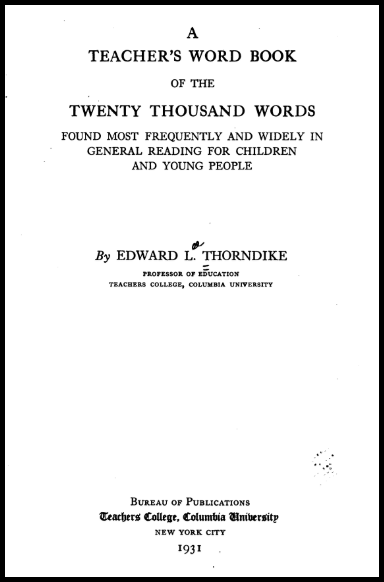
# Глава 1: Путь к веб-корпусу

## 1.1. Этапы становления и развития корпусной лингвистики

На вопрос о том, что такое, собственно, «корпус», есть несколько ответов. Пожалуй, одно из самых широких определений дает словарь Merriam-Webster: *«коллекция записанных высказываний, использующаяся для описательного анализа языка»*. Схожие определения дают Оксфордский словарь («*коллекция письменных текстов, особенно определенного автора или на определенную тему»*), исследователи Тюменского государственного университета *(«набор лингвистических данных из определённого языка в форме записанных высказываний или письменных текстов, доступный для анализа»*). Общее у всех этих определений одно: корпусом может быть любой сборник текстов или высказываний.

Если отталкиваться от такого определения, то получается, что корпусная лингвистика, как и понятие корпуса вообще, уходит корнями в глубины истории. Если взять более близкие к нашему времени исторические периоды, то мы можем обратить внимание на использование эмпирических данных в лингвистике XIX века. Так, употребление слов изучалось и исследовалось на примере цитат из литературы в уже упомянутом выше Оксфордском словаре. Грамматика немецкого языка Г. Пауля широко опиралась на произведения немецких классиков; каждое фонологическое, морфологическое или синтаксическое положение так или иначе опиралось на какие-либо их произведения и подкреплялось цитатами из них. Другой немецкий языковед, V. Käding, составил в 1897 году сборник из 11 миллионов слов (цифра внушительная даже по нынешним меркам!), на основе которого изучал дистрибуцию букв и их сочетаний в немецком письменном языке.

Нельзя также забывать и об использовании корпусных методов в обучении иностранным языкам. Еще с начала XX века американские и британские исследователи начинают собирать языковые данные из учебников, детской литературы и других источников, на основе которых создаются частотные словари, служившие для помощи в изучении английского. Из этих исследователей самым заметным был, пожалуй, Edward Lee, собравший для своей работы *A Teacher's Word Book of the Twenty Thousand Words Found Most Frequently and Widely in General Reading for Children and Young People* корпус из 5 миллионов слов.



***Рис. 1. Один из первых «протокорпусов»***

Однако при всем этом даже самые большие “протокорпуса”, собиравшиеся в тридцатых и сороковых годах, при всем своем многообразии не достигали объема, достаточного для адекватного отображения хотя бы небольшой части языка, а их использование в науке не было в нужной мере систематизировано.Корпусную лингвистику еще ожидал тот скачок, в ходе которого появятся современные корпуса - массивные, организованные сборники языковой информации, которые мы знаем сегодня и без которых современная лингвистика практически немыслима.

«Опорой» для этого скачка было развитие компьютерной техники. Возможность автоматически собирать, скомпоновывать, а затем анализировать тексты позволила лингвистам оперировать большими объемами языковых данных и в полной мере использовать их (при этом компьютер еще и оставался беспристрастным). Именно поэтому большая часть современных ученых определяют корпуса несколько уже, чем в тех определениях, что были рассмотрены в начале главы, понимая под корпусами лишь *компьютеризированные* сборники языковых данных.

Первой весточкой грядущих перемен стала работа Index Thomisticus, написанная священником Roberto Buza в 1951 и посвященная стихам Фомы Аквинского. В ней впервые на основе корпуса был построен автоматически (еще не на ЭВМ, с помощью перфокарт) сгенерированный конкорданс. Через десять лет появился уже первый полноценный компьютеризированный языковой корпус - Брауновский корпус - включавший 500 текстов из американских газет, журналов и книг, по 2000 словоупотреблений каждый. Созданный для системного изучения английского языка, Брауновский корпус также содержал частотный и алфавитно-частотный словари, различные статистические распределения. За этими первопроходцами последовали корпус Ланкастер-Осло-Берген британского английского в 1961 году, Колхапурский корпус индийского английского в 1978 году, Ланкастерский корпус китайского-путунхуа в 1991 году и многие другие.

Разумеется, активный рост корпусной лингвистики не был гладким: некоторые представители научного сообщества восприняли её в штыки и отказывались принимать как достойную изучения дисциплину. В частности, против использования корпусов высказывался такой видный деятель в лингвистике ХХ века, как Ноам Хомский, в основном из-за того, что корпуса использовались для изучения языкового производства (language performance), а не языковой компетенции (language competence), исследование которой (путем интроспекции, а не с помощью анализа уже построенных высказываний) Хомский считал гораздо более важной задачей. “Корпусная лингвистика не значит ничего,” - сказал он в своем интервью в 1997 году, “...Представьте, что физики и химики решат, что вместо того, чтобы опираться на эксперименты, они будут записывать на видео все происходящее в мире.. ..Наука не занимается таким.”

Конечно, Хомский лукавил. Во-первых, науки, работающие почти исключительно с больших размеров “корпусами” данных, уже существовали - астрономия и геология. Во-вторых, корпуса не обязательно должны быть основным инструментом в исследовании языка: они вполне эффективны в качестве вспомогательного средства, имеющего репрезентативную роль, источника той информации, которую не может дать интроспекция (например, частотных данных). В дальнейшем корпусные исследования не только дали науке новые знания, недоступные интроспективным методам Хомского, но и прямо противоречили его интроспективным данным. Так, исследования Т. МакЭмери и Э. Уилсона опровергли *интуитивное* предположение Хомского, что английский глагол perform не может брать в качестве объекта неисчисляемые существительные, обнаружив в корпусных данных часто встречающиеся коллокации *perform magic, perform labour.*

В таком ключе, собственно, и продолжила развиваться корпусная лингвистика. За первыми, еще относительно маленькими корпусами последовали массивные национальные корпуса, ставившие своей задачей в полной мере отобразить все богатство национального языка - Американский национальный корпус в 1990 году, Британский национальный корпус в 1994 году, Национальный корпус русского языка в 2004 году.

Последним крупным скачком для корпусной лингвистики стало развитие всемирной сети, начавшееся в конце прошлого века: здесь первопроходцем стал британский лингвист Адам Килгарифф, написавший в 2001 году статью “Web as Corpus”, в которой онуказал на почти безграничные возможности Интернета для нужд корпусной лингвистики. “Теперь у нас есть всемирная сеть,” - пишет он - “способная дать нам доступ к колоссальным объемам текста, скольких угодно жанров и форм, по одному нажатию кнопки и бесплатно.” (Kilgarriff 2001: 344)

*Рисунок 2: Адам Килгарифф*

Корпусная лингвистика может использовать веб по-разному: с его помощью возможно искать и собирать нужные данные, или же использовать веб-тексты непосредственно вместо корпусов, например, для нужд перевода. Кроме того, сам язык сети Интернет, со всей его необычностью и богатством, представляет немалый интерес для исследователей в области корпусной лингвистики. Конец 2000-х годов и начало следующего десятилетия ознаменовались появлением большого количества “веб-корпусов” вроде семейства корпусов TenTen, семейства корпусов Aranea и Генерального Интернет-корпуса Русского Языка (*ГИКРЯ*).

## 1.2. Развитие программно-лингвистического инструментария

Итак, уже к 1990-м годах корпусная лингвистика окончательно оформляется как научное направление - всего через 40 лет после того, как появились первые современные корпуса - а к настоящему времени представляет собой одну из ведущих дисциплин языкознания вообще. Однако с успехами появились новые проблемы. С развитием корпусной лингвистики начали увеличиваться и объемы получаемых ею данных, размеры самих корпусов: если Брауновский корпус содержал в себе всего лишь миллион словоупотреблений, то Британский национальный корпус - уже сто миллионов. Если же говорить о веб-корпусах, то пределы их роста практически безграничны, как безграничен сам Интернет: англоязычная ветвь TenTen, enTenTen (один из более тридцати таких же веб-корпусов!), начинала с двух *миллиардов* словоупотреблений в 2008 году, а на данный момент приближается к отметке в *пятнадцать* миллиардов.

С одной стороны, чем больше языковых данных, тем лучше они представляют собой язык. С другой стороны, такие большие данные следует каким-то образом резюмировать, приводить их в вид, в котором их легко сможет воспринять человек. Из-за этого наряду со средствами собственно по построению корпусов начинают играть большую роль инструменты для обработки и анализа языковой информации.

В частности, в среде корпусной лингвистики возникает необходимость в точном измерении сочетаемости слов между собой на основе взятых из корпусов статистических конкордансных данных. Хотя, разумеется, статистический подход является лишь одним из нескольких, применяемых в этой области, наряду с лексическими и синтаксическими методами, все же сложно не замечать пользу, которую вносят в неё сухие цифры. Точно так же нельзя переоценить важность изучения сочетаемости в принципе: связи между словами и словосочетания зачастую не находят полного отражения в словарях и в научной литературе, и новыми данными о них смогут воспользоваться как лингвисты-теоретики, так и лексикографы.

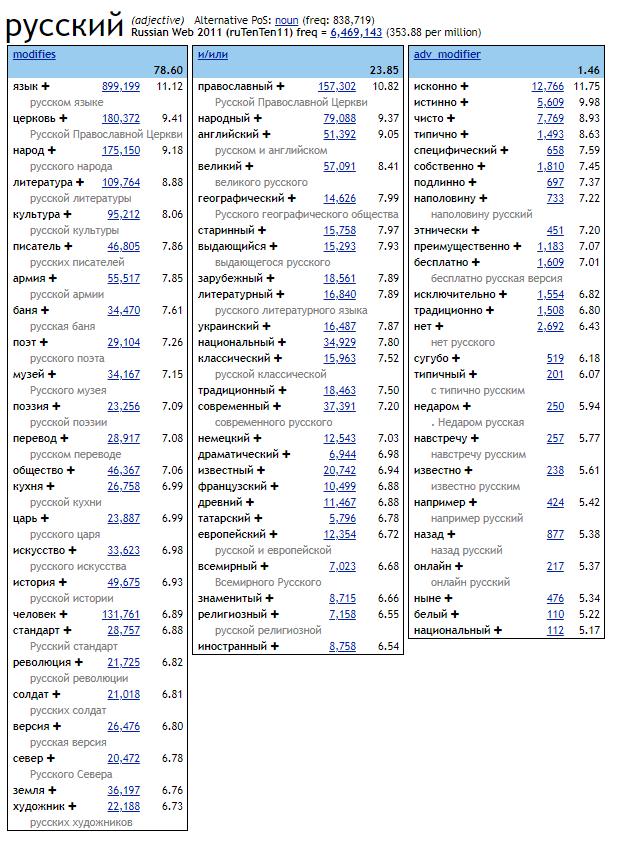
В 1989 году исследователи Ken Church и Patrick Hanks предложили меру Mutual Information, измеряющую то, насколько значима (salient) связь между двумя словами. Эта мера основывается на основе данных о частоте встречаемости в корпусе коллокаций (словосочетаний) и о частоте встречаемости образующих их коллокатов. Это новое веяние в лингвостатистике нашло широкое применение корпусной лингвистике, а также породило, по сути, новую область в языкознании: лексическая статистика.

Разумеется, решение Черча и Хэнкса не было идеальным. Их мера отдавала предпочтение редким словам, а списки слов были основаны на словоформах (*человек, человека, людей*), а не на леммах (*человек*). Эти две проблемы решались довольно просто – созданием корректирующих статистику весов и лемматизаторов соответственно. Однако на них сложности не оканчивались. Во-первых, точность измерений сильно ухудшал шум в данных. Во-вторых, неопределенным был размер самого контекстного окна, который мог быть как и симметричным (то есть с одинаковым количеством коллокатов перед и после слова), так и ассиметричным. Наконец, мера не учитывала, как именно слово было связано со своим контекстом: соседнее слово могло быть определением, обстоятельством, объектом или субъектом, либо же вообще не иметь с ним грамматической связи. Хотя от этих проблем страдали и другие меры вычисления синтагматической связанности, все же.

Все эти вопросы были решены уже в следующем столетии британским лингвистом Адамом Килгариффом (Adam Kilgarriff), предложившим новый формат выдачи конкордансных данных из корпусов. Это так называемый wordsketch: автоматически строящаяся страница, отображающая грамматическое и коллокационное поведение слова. Для создания скетча необходимы четыре вещи: корпус, средства NLP (токенизатор, лемматизатор и т.д.), скетч-грамматика для языка и статистические данные для того, чтобы выбирать значимые коллокации.

Революционным в этом формате было то, что в нем рассматриваются грамматические отношения между словами в предложения: вместо того, чтобы брать фиксированный контекст длиной n1 слева от искомого слова и n2 справа от него, «скетч» рассматривает его грамматические отношения. В ранней версии созданной Килгариффом системы, опубликованной им в 2004 году, различались 27 вида грамматических отношения для английского языка и 23 – для чешского. Так, глагол мог связываться с субъектом, объектом, другим глаголом (образуя сложное сказуемое), обстоятельством, с предлогами и местоимениями, причем каждый тип связи отделялся в отдельный список, и эти списки могли рассматриваться, в зависимости от нужд лексикографа, либо как единое целое, либо поодиночке. На рисунке 3 ниже мы можем увидеть скетч существительного «русский», созданного на основе корпуса *RussianWeb 2011* (ruTenTen11).

Новый формат, предложенный Килгариффом, получил широкое признание за свою эффективность, и его изыскания в конечном итоге привели его к созданию целого веб-ресурса, доступного сейчас каждому пользователю сети Интернет – SketchEngine. Эта сетевая платформа, одна из самых успешных в своем роде, служит для сбора, построения и исследования веб-корпусов, при этом «скетчи» образуют ее центральное ядро. Она не только содержит более 400 корпусов на более чем 80 языках, но также дает пользователю возможность строить корпуса самостоятельно (автоматически, на основе материала из интернет-страниц). Система может не только просто показывать «скетчи», но и сравнивать их, показывая общие и расходящиеся грамматические связи, а также “дистрибуционный тезаурус”, позволящий отображать слова со схожими грамматическими связями в виде облака (word cloud).



*Рисунок 3: Скетч существительного «русский»*

За свое существование система SketchEngine пережила заметный рост и нашла применение в самых разных областях языкознания. Первыми, кто начал использовать её, стали лексикографы: к 2014 году SketchEngine использовали четыре из пяти крупнейших издателей словарей в Великобритании (Cambridge University Press, Harper Collins, Macmillan и Oxford University Press), а также целый ряд национальных институтов, занимающихся словарной работой (в Ирландии, Словакии, Болгарии, Чехии, Голландии, Ирландии и Эстонии). В университетской среде она нашла применение в изучении дискурсного анализа и NLP-исследованиях; высокую эффективность она показала также в обучении языку, для которого Килгариффом был разработан отдельный вспомогательный инструмент, SkELL (Sketch Engine for Language Learning). Кроме того, SketchEngine активно используется переводчиками, терминологами и компаниями, занимающиеся языковыми технологиями.

# Глава 2: Работа системы SketchEngine

## 2.1. Предобработка

### 2.1.1. Построение корпуса в SketchEngne

Для того, чтобы понимать процесс предобработки корпусов для нашей системы, разумно сначала рассмотреть, какими способами производится построение личных корпусов в SketchEngine. В этих целях в рамках сайта существуют два основных метода разной степени сложности. Во-первых, мы можем собрать данные для корпуса вручную, так, как это делалось в корпусах ХХ века. В таком случае мы выбираем либо файлы непосредственно на нашем компьютере, либо ссылки на файлы в Интернете. Для нас основным плюсом ручного способа будет то, что мы сможем обработать данные для корпусов еще до того, как предобработку за нас сделает сам SketchEngine.

Сайтом принимаются файлы разных форматов, включая.doc, .docx, .htm, .html, .pdf, .ps, .tmx, .txt, .vert, .xml., а также архивы .zip, .tar, .tar.gz, и .tar. bz2. Однако здесь существует несколько нюансов. Файлы формата PDF, как вполне ожидаемо, в процессе конвертирования в чистый текст могут подвергнуться сильному искажению, если в них присутствуют колонки, тексты в несколько столбцов и т.д.; слова могут разрываться или пропадать вовсе. Файлы XML же будут правильно обрабатываться только тогда, когда в нем будет присутствовать корректная маркировка, включающая в себя необходимые метаданные, как например:

*<xml>*

*<doc author="Jan" title="Example doc 1">*

*<p>This is a paragraph.</p>*

*<p>This is another paragraph.</p></doc>*

*<doc author="Jan" title="Example doc 2">*

*<p>I will add some more text here.</p>*

*</doc>*

*</xml>*

Другой вариант создания корпуса отсылает нас к уже упомянутым словам создателя SketchEngine: на нашей стороне Всемирная сеть, которая дает нам собирать данные огромных размеров почти мгновенно и при этом бесплатно. В этом случае задействуется служба WebBootCaT, которая производит самостоятельный поиск по Интернету, используя поисковую систему Bing. В качестве исходных данных возможно использовать либо определенные слова-ключи (seed words), в случае чего система произведет определенное количество поисковых запросов на каждое слово-ключ, самостоятельно забирая определенное количество страниц с каждого запроса.

Процесс происходит полностью автоматически, однако пользователь, тем не менее, в состоянии так или иначе управлять им, задавая количество запросов, длину цепочек, количество страниц, заносящихся в корпус с каждого запроса, а также минимальный и максимальный размер страницы. Кроме того, мы можем опускать определенные сайты, расширения и слова, а также включать в поиск ключевые слова, которые должны будут находиться во всех нужных нам страницах (тэговые слова ищутся цепочками и, как следствие, некоторые из них могут опускаться). Таким образом мы получаем достаточно широкий спектр возможностей, который позволяет нам довольно свободно искать нужные нам данные.

Схожим способом создаются "подкорпуса" - еще один, вспомогательный способ построения корпусов в SketchEngine. В данном случае также выделяются определенные ключевые слова или страницы, на основе которых будет строиться корпус, однако вместо поиска по Интернету система будет искать в пределах одного из уже имеющихся корпусов, "вырезая" из него определенную часть и расссматривая ее отдельно.

Построение корпуса самостоятельно, когда исследователь самостоятельно выбирает страницы, можно сравнить с работой средневекового мастера - ручной, тщательной, малоэффективной, но зато позволяющей лучше подогнать итоговое изделие под конкретного человека или задачу. WebBootCat же напоминает по своей эффективности промышленное производство: корпус собирается гораздо быстрее, при этом процесс нахождения нужных страниц значительно упрощается. Однако при этом возможности предобработки при ручной загрузке текстов и при использовании WebBootCat различаются. В первом случае мы можем производить очистку текстов средствами внешних инструментов, так как искомые веб-страницы можно сохранить на компьютере и там уже модифицировать на свое усмотрение. Если же мы используем автоматические методы сборки корпуса, то все, что мы можем сделать – это убрать перед сборкой корпуса страницы, которые мы найдем заведомо неправильными.

В рамках этой работы будут рассматриваться только корпуса, построенные с использованием WebBootCat: методы обработки текстов до того, как они попадут собственно в корпус, рассматриваться не будут. Во-первых, эти методы обычно достаточно трудоемки и занимают много времени, из-за чего у нас просто не остается, собственно, времени разобрать их полностью.. Во-вторых, они связаны больше непосредственно с «очисткой» текста, и собственно с темой корпусной лингвистики соприкасаются мало. Однако проблемы предобработки, поднимаемые в этом вопросе, точно так же относятся и к собранным вручную корпусам.

### 2.1.2. Проблемы с лемматизацией

В корпусах SketchEngine достаточно часто мы видим слова, которые не лемматизируются либо лемматизируются неправильно, либо из-за ошибок самого лемматизатора, либо из-за плохого качества текста(что характерно для веб-корпусов). Ошибки, встречающиеся в пользовательских корпусах, достаточно разнообразны, однако рассмотрев корпуса русского языка, возможно выделить самые распространенные их типы.



Для этого с помощью WebBootCat были созданы два корпуса: один по ключевым словам, связанным с Китаем *("Китай", "китайцы", "китайский"),* и другой по словам, связанным с Таиландом *("тайский", "тайцы", "Таиланд").* Специальный алгоритм проверял все леммы, взятые из этих корпусов, на наличие в Викисловаре, оставляя только те, которые не присутствовали в последнем (чаще всего это происходило из-за того, что слово были неверно приведено в словарную форму, либо же этого не случилось вовсе).

*Рисунок 4: Ошибка в лемматизации. «Цзиньпина» вместо «Цзиньпин.*

Как можно было ожидать, чаще всего лемматизатор дает сбой на именах собственных. В частности, часто не лемматизируются названия стран, городов, межгосударственных организаций: при этом достаточно странно то, что система "спотыкается" не только на довольно малоиспользуемых топонимах (*"Бутаном"* вместо *"бутан"*), но и на достаточно распространенных терминах (*"Евросоюза"* вместо *"евросоюз"*, *"Монголией"* вместо *"монголия").* Схожая ситуация происходит с именами собственными (*«Цзиньпина»* вместо *«Цзиньпин»*)*.*

Ошибки стабильно совершались в случае тех слов, которые, будучи в целом общепринятыми, не являются при этом официальными, "правильными". Помимо уже упомянутого "Евросоюза" (при своей популярности это сокращение редко используется в официальных текстах), лемматизатор не мог правильно привести к словарной форме распространенное ошибочное написание слова *"Таиланд" - "Та****й****ланд".*

Еще одной крупной проблемной областью стали слова, включающие в себя дефисы. Такие слова практически полностью не лемматизировались корректно: иногда они считались одним словом, хотя по смыслу представляли собой скорее словосочетания (например, *"город-мегаполис", "Москва-Пеки*н"). Иногда они, наоборот, разделялись, из-за чего дефектная лемма начиналась, собственно, с дефиса (в частности, сокращение "С.-Петербург" трактовалось как две отдельные леммы - "с." и "-петербург". В дополнение к этому даже те слова с дефисом, которые не были разделены (и не должны были быть разделены по смыслу), чаще всего не приводились к правильной словарной форме *("воздушно-десантные*" вместо *"воздушно-десантный").*

Среди более мелких проблем можно выделить как и вполне очевидную неверную лемматизацию каких-то редких, специфичных слов (хотя странным оказалось то, что система спотыкалась даже на таких, казалось бы, простых словах, как "автодорога", "колледж", "престижный"), а также оказавшиеся неожиданными случаи ошибок с лемматизацией в случае слов, написанных заглавными буквами (*"ВЛЕЗАЙТЕ", "ДРАКИ", "КИТАЙЦАМИ"* вместо ожидаемых *"влезать", "драка", "китаец").*

Более глубоко ошибки лемматизации будут исследованы в следующей главе, где будут использоваться непосредственно статистические данные и на их основе делаться определенные выводы; пока что мы просто отметим, что эти ошибки просто-напросто существуют и приносят определенный вред для корпуса.

## 2.2. Постобработка и визуализация

### 2.2.1. Методы работы с корпусом.

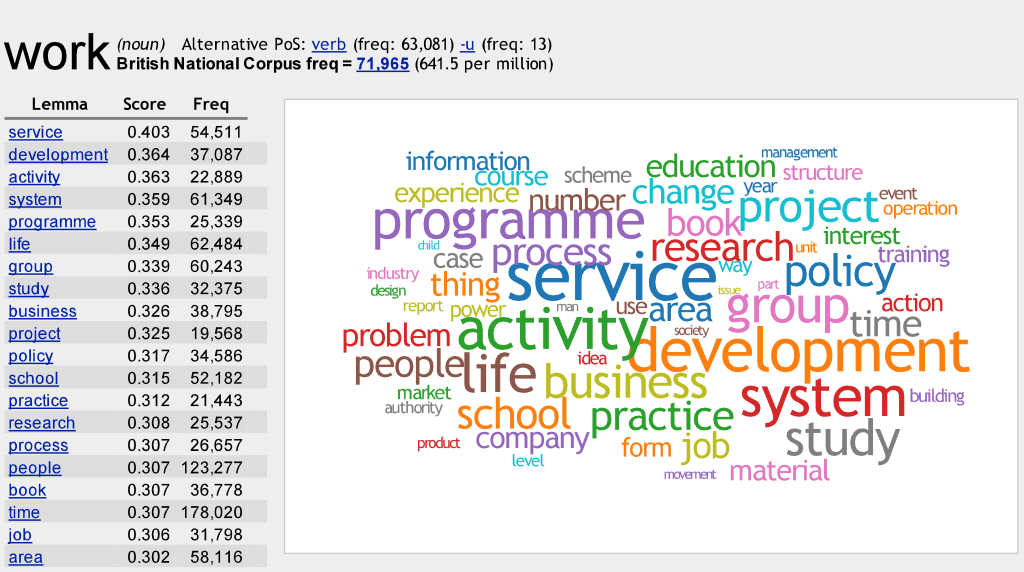
Хотя предобработка корпусов и представляет определенную ценность для научно-исследовательского комплекса, работа в целом отнюдь не ограничивается ею. Даже самый точный и всеобъемлющий корпус не будет иметь никакой пользы, пока не существует методов, позволяющих извлекать из этого корпуса некие полезные для нас статистические данные. К нашему счастью, SketchEngine в этом плане предоставляет достаточно широкий инструментал. Конкретные формулы, которые используются в этом инструментале, будут разобраны чуть позже, сейчас же мы рассмотрим все имеющиеся в системе средства более поверхностно, по области применения.

Хотя SketchEngine дает возможность находить коллокации традиционным способом, основным методом исследования корпусов, который и дал сайту свое название, в нем является "скетч" - таблица «коллокаций в разрезе», показаывающая грамматические отношения между словами. Вместо просто близости двух слов друг к другу, как это происходит при простом поиске коллокаций, "скетч" учитывает также и грамматические отношения между ними, разводя коллокации согласно этим отношениям в отдельные части таблицы. Например, отдельно могут показываться коллокации между глаголом и его дополнением, глаголом и обстоятельством, дополнением и определением и т.д..

Таким образом итоговый результат становится более организованным и легким для восприятия; становится легче делать суждения не только о лексике, но и о синтаксисе языка. "Скетчи" особо полезны для лексикографии, однако находят применение также и в области обучения языку.

Помимо этого основного инструмента, в рамках SketchEngine существует и иные методы исследования корпусов, основанные на коллокациях и скетчах. В частности, при наличии пользовательского корпуса возможно автоматически извлекать ключевые слова и словосочетания ("термины"), свойственные именно этому корпусу. Для этого его коллокации сравниваются с таковыми у другого корпуса: этот корпус служит как стандарт, и отхождение от него в плане частоты, собственно, и является критерием для термина. Например, при сравнении корпуса о Китае, который использовался для проверки ошибок предобработки, с корпусом-мерилом RuTenTen 2011, SketchEngine выдал такие "термины", как "КНР", "Тянанмень", "Храм Неба" и "Запретный город" (однако наряду с ними присутствовали как и сомнительные "Пекин", так и вовсе ошибочные "поднебесный" и "торговых центр", являющиеся результатом неверной лемматизации).

Другим важным средством в инструментарии SketchEngine является построение автоматического тезауруса (здесь слово "тезаурус" обозначает словарь семантически близких слов), основанный на принципе на теории дистрибутивной семантики. Процесс построения тезауруса проходит в два этапа: сначала строится скетч на основе искомого слова, после чего из всех слов той же части речи, имеющихся в корпусе, выбираются те, которые имеют больше всего схожих коллокаций. Предполагается, что слова, появляющиеся в схожем контексте, имеют при этом и схожее значение.



***Рис. 5. Тезаурус SketchEngine в действии. Score обозначает силу семантической связи, Freq – частотность.***

В целом автоматические тезаурусы находят применение там же, где и тезаурусы обычные. За счет специфики процесса построения автоматических тезаурусов в них могут попадать некоторые ненужные слова (например, антонимы), зато они могут покрывать те области языка, которые часто отсутствуют в традиционных словарях синонимов: неологизмы, сложные слова, технические термины и т.д..

### 2.2.2. Меры и формулы.

Как уже было сказано, большая часть инструментария, который используется в SketchEngine, основан, так или иначе, на вычислении коллокаций между словами. Конкретно для них существует несколько способов измерения. Как уже было сказано, до 2006 года основной мерой вычисления коллокаций был метод MI-Score:

После этого этот метод сменился методом AssociationScore (AScore), а затем методом logDice, основанным на «коэфициценте Дайса» и особой гибридной мерой (по причинам, описанным выше).

Помимо этих мер, основанных на грамматических отношениях и использующихся в основном для скетчей, есть иные меры, не учитывающие их и использующиеся только в коллокациях.

* *(T-Score)*
* *(MI3-Score)*
* *(log-likelihood)*
* *(minimum sensitivity)*
* *(MI.log-f)*
* *(Dice)*
* *(relative freq)*

### 2.2.3. Решения проблем

Разумеется, при всей широте возможностей функционал SketchEngine не является идеальным. Его возможности рассчитаны на использование для самых разных корпусов, на самых разных языках и на самые разные темы, из-за чего система не в полной мере задействует возможности конкретных языков и меньше сосредотачивается на конкретных задачах. К счастью для нас, сайт SketchEngine имеет развитый API, позволяющий использовать его возможности для более специализированных задач вроде нашего комплекса.

Следует начать с того, что несмотря на то (вернее сказать, потому), что система SketchEngine в основном основана на статистических методах, так или иначе задействующих частотность, из нашего внимания часто пропадают иные способы работы с корпусами, более узкие и более заточенные под конкретные языки или области языка. В частности, очевидно, что нынешний метод извлечения терминов (почти полностью полагающийся на статистические данные и имеющий только одно правило), может работать эффективнее, если будет опираться помимо частотности также на данные из терминологических словарей. Полученные системой термины можно было бы пропускать через эти словари, удаляя (или снижая релевантность) тех слов и словосочетаний, которые в них отсутствуют. В журнале *Research in Computing Science* 70 2013 года говорится о эффективности использования терминологических словарей в области биомедицины. По сути, необходимое условие здесь только одно: в нашем языке должны иметься соответствующие словари в свободном доступе для соответствующих областей науки.

Кроме того, мы можем задаться еще одним вопросом: насколько эффективны статистические методы сами по себе? Формулы, используемые в инструментах SketchEngine, обычно задействуют только одну меру, хотя для коллокаций и возможно посмотреть сразу несколько мер параллельно. Вполне возможно измерять коллокации по "пучкам" мер, например, по хорошо соотносящимися друг с другом тройками.

Кроме улучшения функциональности уже имеющихся инструментов, в наших силах также еще и расширять возможности инструментария SketchEngine, создавая новые способы работы с коллокациями. В частности, нынешний тезаурус, используемый в SketchEngine, может описывать синонимический фон только одного слова, что является большим ограничением для его эффективности.

# Глава 3: Рабочий процесс

## 3.1. Предобработка

Как и теоретическая часть работы, практическая часть разделена на три основные части: предобработка, постобработка и визуализация. В первую область входит рассмотрение различных связанных с предобработкой ошибок и предложение каких-то решений для них (собственно решения, к сожалению, слишком сложны и разноплановы для включения в эту работу). В частности, здесь мы рассмотрим самые распространенные ошибки при лемматизации на разных персональных корпусах. Помимо корпуса о Китае, о котором мы уже говорили, были также построены и исследованы два других персональных корпуса: о Германии и о медицине.

При исследовании выборки из 5000 лемм, взятых из корпуса слов, связанных с Китаем, 2,5% оказались дефектными. Из этих слов 20% - этнонимы и топонимы, 17% являются составными и содержат дефисы, с 9% лемматизатор дал сбой, так как слово было полностью либо частично записано заглавными буквами, еще 9% представляли собой слова из HTML-«начинки» страниц. 3% лемм не были обработаны по более простой причине – они были написаны с грамматической ошибкой или опечаткой. Остальные 42% лемм не были верно лемматизированы по каким-то иным причинам – либо это были редкие слова (часто термины), либо неологизмы, в литературном языке использующиеся редко и, вероятно, именно по этой причине не опознанные системой («шопинг»).

Схожую, но иную картину мы получаем, когда рассматриваем выборку с корпуса, основанного на медицинских терминах. Здесь немного больше сам процент необработанных лемм – 3% вместо 2% - при этом составных слов, слов из «начинки», топонимов и слов в этой выборке оказалось значительно меньше – 6,6%, 10%, 6% и 2% соответственно. Вместо этого среди неверно обработанных лемм оказалось очень много терминов: названия болезней, лекарств и т.д. Это весьма важно для нас, так как показывает нам то, что при создании терминологических корпусов «нужные» термины часто будут выпадать.

Корпус, посвященный Германии, дал нам результаты, схожие с корпусом о Китае, но с определенными нюансами. В этом корпусе изобиловали топонимы и этнонимы, которые составляли из себя примерно треть из всех неправильно лемматизированных слов. 11% неверных лемм представляли собой составные слова, около процента – слова из «начинки».

На основе полученных данных мы можем определить своего рода «группы риска» - те слова, с которыми лемматизация чаще всего делает ошибки. В первую очередь это оказались даже не составные слова, которые встречаются не во всех текстах, а малоупотребляемые термины типа «антрополог» или «полисахариды», а также топонимы, этнонимы и другие имена собственные. В рамках этой работы мы ограничимся лишь определением этих наиболее вероятных ошибок; о том, как в будущем можно будет бороться с этими ошибками, будет сказано в конце работы.

## 3.2. Постобработка

### 3.2.1. Извлечение терминов

Часть комплекса, связанная с постобработкой текста, имеет более разноплановый характер. В общем и целом ее можно разделить на три части: улучшение работы имеющихся инструментов, объединение необходимых для инструмента мер и расширение их возможностей. В рамках работы каждая из этих областей будет рассмотрена на примере одного из инструментов системы SketchEngine: извлечения терминов, построения тезаурусов и поиска коллокаций. В дальнейшем комплекс будет расширяться, и впервые рассмотренные здесь методы будут использоваться и для других инструментов системы.

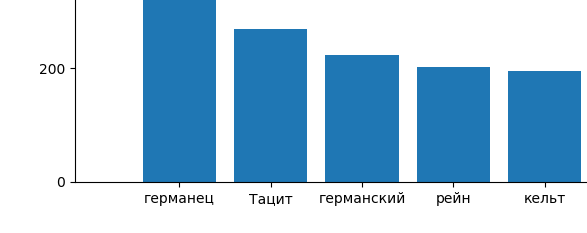
В области извлечения терминов, как уже было сказано, для улучшения работы инструмента используются терминологические словари, которые покрывают те или иные области языка, и данные из которых могут служить для отделения верных терминов от неверных. Таким образом будет точнее определяться тема корпуса и составляющих его текстов. За лексикографическими источниками на данном этапе мы обратимся к разным источникам. Сайт "Грамота.ру", который содержит в себе ссылки на несколько крупных терминологических словарей. Среди этих словарей можно найти анатомический, биологический, военный словари, "православную библиотеку", словарь гендерных, маркетинговых, шрифтовых терминов. Еще больше информации можно почерпнуть из Викисловаря, в котором имеется 117 терминологических категорий, от спелеологии до рекламы. Достаточно сложная система, основанная на семантических связях, действует для сайта-словаря glossary.ru. Как мы видим, лексикографическими данными русский язык не обделен.

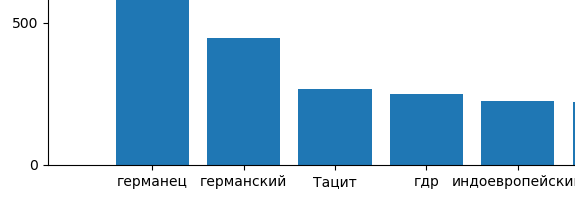
На первый взгляд может показаться, что решение здесь элементарно: мы просто берем ключевые слова и словосочетания, полученные SketchEngine непосредственно из полученных корпусных данных, после чего удаляем все, что не находится ни в одном из терминологических словарей. Однако здесь существует несколько нюансов, которые необходимо учесть. Во-первых, следует помнить, для чего существует инструмент извлечения терминов в первую очередь. По сути, корпусные термины определяют тему корпуса в целом: в идеале они должны принадлежать к одному или хотя бы к небольшому количеству словарей. Во-вторых, ключевые слова не обязательно должны быть терминами в прямом смысле этого слова. К политике могут иметь отношения имена политиков и названия стран; к юриспруденции - "феня" и иная преступная лексика. Поэтому вместо того, чтобы удалять все не входящие в словари ключевые слова, гораздо эффективнее было бы просто взвешивать их.

В итоге процесс взвешенного извлечения терминов выглядит так. Первый этап - выделение искомых тем. Это можно проделать как и вручную, так и автоматически (выбираем те словари, слов из которых больше всего среди полученных SketchEngine терминов). После этого проводится взвешивание "сырых" терминологических данных. Самые высокие весы - 2 - получают слова, присутствующие в словаре (или словарях) по основной теме текста. Аналогично весы 1,5 получают те термины, которые относятся к двум "второстепенным" темам. Если слово не присутствует ни в одном из словарей, то его коэффициент "термовости" не меняется, если находится в хотя бы одном из словарей - опускается вполовину (весы 0,5).

Для создания собственно словарей использовались терминологические категории Викисловаря.

Алгоритм взвешенного извлечения терминов был проверен на «китайском», «медицинском» и «немецком» корпусах, ранее использовавшихся для исследования предобработки (в качестве словаря-мерила использовался ruTenTen11). Эффективность этого алгоритма сильно варьировалась от корпуса к корпусу. Например, в «немецком» корпусе присутствовали нерелевантные для Германии ключевые слова «Тацит» и «кельт», которые появились там из исторического текста. Они пропали со второго и четвертого мест, спустившись на третье (не очень впечатляет) и тридцать третье (впечатляет весьма и весьма) места соответственно. В это же время более немецкий «термин» «ГДР» поднялся с пятнадцатого места на четвертое. В «медицинском» корпусе названия болезней поднялись выше имен врачей, их изучавших. Однако в «китайском» корпусе весы не имели большого успеха: положение основных ключевых слов не изменилось, так как они представляли собой заимствования из китайского языка («хуацзяо», «хуажень»), в русских словарях отсутствующие.





***Рис. 6. Диаграммы, показывающие «силу» ключевых слов до и после процесса взвешивания. Тацит теперь почти наравне с ГДР.***

### 3.2.2. Создание мер

Как уже было сказано, большую помощь для создания более точных методов сравнения коллокаций могут оказать «смешанные» меры, совмещающие в себя несколько способов измерения коллокации. Для этого нам придут на помощь различные методы измерения корреляций. Здесь основной задачей для нас является обнаружение крупных трендов, которые встречаются в разных мерах: мы найдем схожести и различия между ними и объединим похожие вместе, что даст нам возможность обходиться не множеством, а всего лишь двумя-тремя из них.

Основных методов для вычисления корреляций два: коэффициент Пирсона и коэффициент Спирмена. Первый служит для вычисления исключительно линейных отношений между двумя переменными (например, массы и веса), второй же оценивает то, насколько хорошо можно описать отношение между двумя переменными с помощью монотонных функций вообще. Для наших целей более полезен коэффициент Спирмена ("чистой" линейной корреляции между мерами добиться сложно, если не вовсе невозможно), и именно его мы использовали для создания смешанных мер.

Процесс создания смешанных мер проходил в три этапа. Сначала для каждой из возможных троек-*M* (сочетаний мер,, ) - вычисляется уровень корреляции (средний коэффициент Спирмена *c* между тремя парами мер в составе тройки – ). Пять троек с самой сильной средней связью выделяются, после чего для каждого элемента в тройку вычисляется также и свой коэффициент, основанный на средней корреляции с двумя другими элементами (по формуле; аналогично с , ). После того, как эти переменные были вычислены, значения разных мер соединяются по пяти самым связным тройкам, после чего меры в них, во-первых, нормализуются (чтобы меры с большим абсолютным значением в целом не затмевали все остальные), а во-вторых, умножаются на весы, равные вышеупомянутому персональному коэффициенту. В итоге мы снова получаем список слов и коэффициентов коллокаций, только этих коэффициентов теперь остается только пять.

При использовании этого алгоритма при поиске леммы «человек» в словаре ruTenTen11 мы получили такие тройки. В целом можно примерно осмыслить полученный в итоге результат: похоже, что тройка T-score, loglikelihood и MI3 выдает в основном часто используемые слова, а остальные тройки – характерные именно для этого слова. Что интересно, и там, и там большую роль играют служебные части речи, особенно предлог «у».

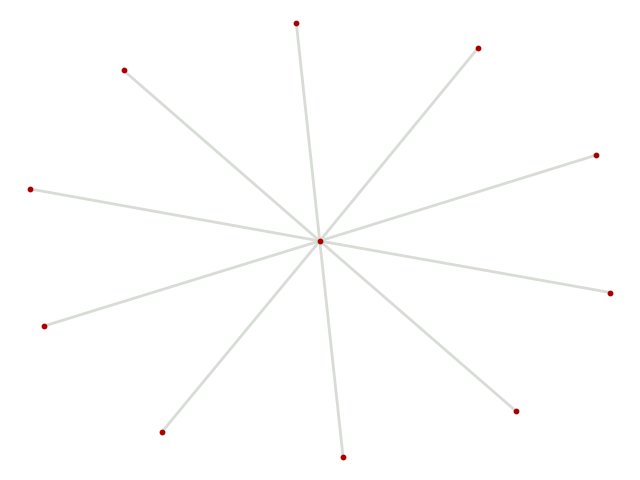
Иные результаты были достигнуты при исследовании

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **'MI.log\_f'**  **'logDice'**  **'min. sensitivity'** | **‘MI.log\_f'**  **'MI'**  **'min. sensitivity'** | **'T-score', 'loglikelihood', 'MI3'):** | **'logDice', 'min. sensitivity'** | **'MI.log\_f', 'MI', 'logDice'** |
| *молодой* | *молодой* | *и* | *тысяча* | *молодой* |
| *тысяча* | *тысяча* | *молодой* | *у* | *тысяча* |
| *у* | *право* | *с* | *другой* | *право* |
| *другой* | *у* | *что* | *право* | *жизнь* |
| *право* | *жизнь* | *в* | *каждый* | *у* |

### 3.2.3. Расширенные тезаурусы

Наконец, последним этапом во всем комплексе является создание новых возможностей для уже существующих инструментов. В рамках нынешнего тезауруса SketchEngine возможно найти семантически близкие слова лишь для одного слова. Из-за этого его возможности весьма ограничены: мы можем узнать только о непосредственных «соседях» слова по семантическому контексту, но не о том, каких «соседей», в свою очередь, имеют они.

Однако если мы проведем операцию построения тезауруса дважды – сначала для центрального слова, а затем для всех его «соседей», то мы получим целую синонимическую сеть, своего рода семантическое поле с центром в виде одного начального слова. В этой сети, в свою очередь сможем выделить семантический «центр» - то есть слова, имеющие много смысловых связей со своими «соседями» - и «периферию», отстоящую по смыслу от этого центра достаточно далеко. Таким образом работает сайт Thinkamp Visual Thesaurus, предоставляющий для каждого слова граф с близкими по смыслу словами в качестве вершин.



***Рис. 7. Графы, показывающие обычный и расширенный тезаурусы***

## 3.3. Визуализация

### 3.3.1. Принципы визуализации

От графов мы плавно переходим к конечной части нашей работы - визуализации обработанных данных, полученных после работы нашего комплекса над полученными из SketchEngine данными.

При ней мы должны следовать определенным принципам дизайна, чтобы итоговый результат был достаточно наглядным и приятным для глаз. Исследователь Steven Few описал некоторые такие правила (Few, 2006):

1. **Визуальная ясность** (visual clarity). Следует избегать большого скопления фигур и неоправданного использования ярких цветов, убирать ненужную информацию и выделять нужную. Таким образом когнитивная нагрузка на смотрящего будут уменьшена, а его способность воспринимать информацию – улучшена.
2. **Ясность данных** (contentclarity). Надписи и легенды не должны быть удалены без надобности. Даже если какие-то данные не указаны, на них должна иметься ссылка. Разумеется, первые два принципа конфликтуют, что приводит к проблемам.
3. **Наглядность** (preattentiveness): такие визуальные свойства, которые человек может воспринять за короткий срок, не просматривая через изображение несколько раз. Например, легко найти синий круг среди синих квадратов, либо же красный квадрат между синих квадратов: сами по себе цвет и форма видны с первого взгляда. Добавив два различия, однако, мы не добьемся позитивного эффекта – чтобы отделить из всего вышеперечисленного красный круг, требуется посмотреть на изображение как минимум дважды.
4. **Гештальт-принципы**. Объекты, близкие друг к другу пространственно (принцип близости) или имеющие схожий вид (принцип схожести) будут в основном рассматриваться как принадлежащим к одной и той же группе. (Пример: изображения кнопок в интерфейсе находятся близко.)
5. **Интерактивность (программы).** Визуализация может подвергаться изменениям, при которых один из предыдущих принципов выходит на первый план в зависимости от того, какой из них на данный момент актуальнее. Интерактивность может быть простой и заключаться просто в возможности повернуть или приблизить график, либо же быть более сложной и включать в себя кастомизацию графиков и данных.

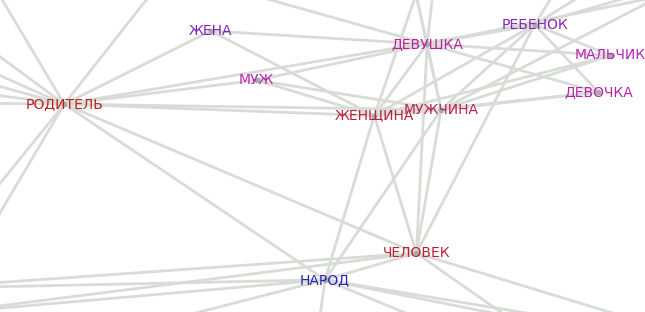
О принципах и проблемах визуализации конкретно корпусных данных писал William L. Allen в своей работе «*Making Corpus Data Visible: Visualising Text With Research Intermediaries»*. Там отмечается, что лингвисты должны показывать “значительные отношения между данными и лингвистикой, но не просто между данными и алгоритмом” (Allen, 2017, с. 12). При своей доступности средства визуализации не всегда достаточно ценятся исследователями, которые должны понимать, что “визуализации и процессы, создающие их, требуют особых ценностей, навыков и целей” (Allen, 2017, с. 12). Аллен приводит еще три принципа, о которых, по его мнению, нельзя забывать при создании визуализаций.

1. учитывать цели и значения до начала работы над дизайном;
2. налаживать четкую связь между всеми членами команды с разными навыками *(для нас не имеет значения);*
3. делать так, чтобы визуализируемые признаки соответствовали целям и дизайну программы.

### 3.3.2. Визуализация расширенного тезауруса

Проблема визуализации особенно остро стоит для построения расширенного тезауруса. "Нативная" визуализация простых тезаурусов в SketchEngine представляет собой довольно примитивное "облако слов"; его минусом, в свою очередь, является то, что несмотря на внешне приятный вид, выражать какие-то величины облако может только размером составляющих его слов, из-за чего больше одной переменной они показать не могут. Тезаурус по типу Thinkmap, представляющий собой граф, вполне адекватен для тезауруса-сети, однако мы должны учитывать, что SketchEngine дает нам не только отношения между словами, но еще и силу этой связи. Кроме того, важно выделить «центр» и «периферию» тезауруса.

В результате мы пришли к двум вариантам рисования графа. В первом варианте позиции узлы и ребра графа расставлены согласно силовому алгоритму Фрухтермана-Рейнгольда и представляют собой несколько пучков близко связанных с собой слов. Так мы можем легче выделять семантические поля внутри текста.



***Рис. 8. Расширенный тезаурус от слова «человек». Первый вариант графа.***

Второй же вариант графа представляет собой нечто среднее между, собственно, графом и пресловутым «облаком слов». Он представляет собой группу увеличивающихся колец овальной формы: слова, имеющие много семантических связей, находятся в центре; имеющие только одну или две – на периферии. Помимо этого, эти слова еще и выражают с помощью своего суммарную силу связи со всеми своими соседями. В первом варианте эта же величина выражается только цветом – от синей периферии через фиолетовый к красному центру.

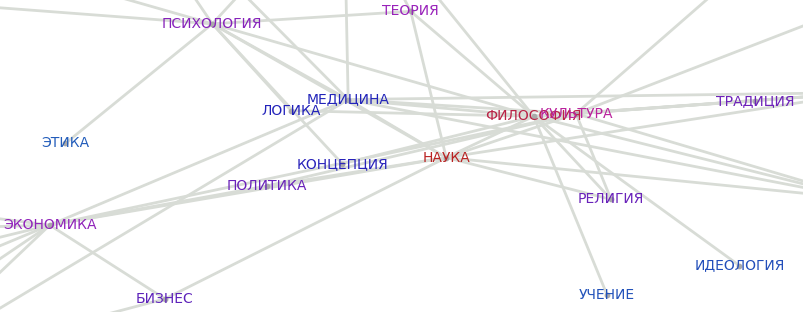


***Рис. 9. Расширенный тезаурус от слова «человек». Второй вариант графа.***

Второй вид графа выглядит куда более эффектно и в некоторой степени более наглядно, зато первый лучше показывает отношения между конкретными словами.

Во время исследования расширенного тезауруса обнаружилось несколько интересных фактов. Во-первых, зачастую реальным центром расширенного тезауруса становится даже не то слово, с которого началось построение графа, а один из его соседей, выражающий более общее понятие и имеющий за счет этого больше сильных семантических связей.

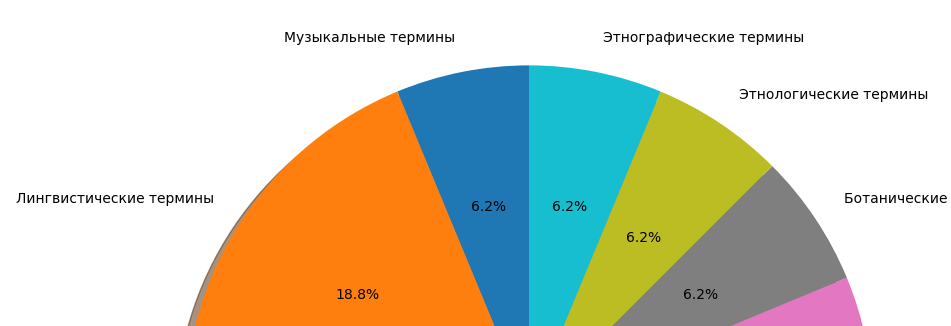
В частности, при попытке построения расширенного тезауруса от слова «медицина» мы пришли к ситуации, где ярко-красным центром стало слово «наука», а ее ближайшими соседями – «религия» и «философия». Медицине же были назначены темно-синий цвет и место рядом с другими науками вроде логики и психологи, близкое к центру, но все же не непосредственно *в* нем.



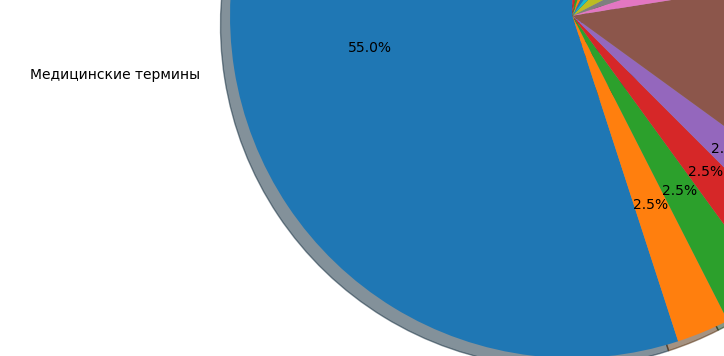
***Рис. 10. Расширенный тезаурус от слова «медицина».***

### §3.3.3. Другие визуализации

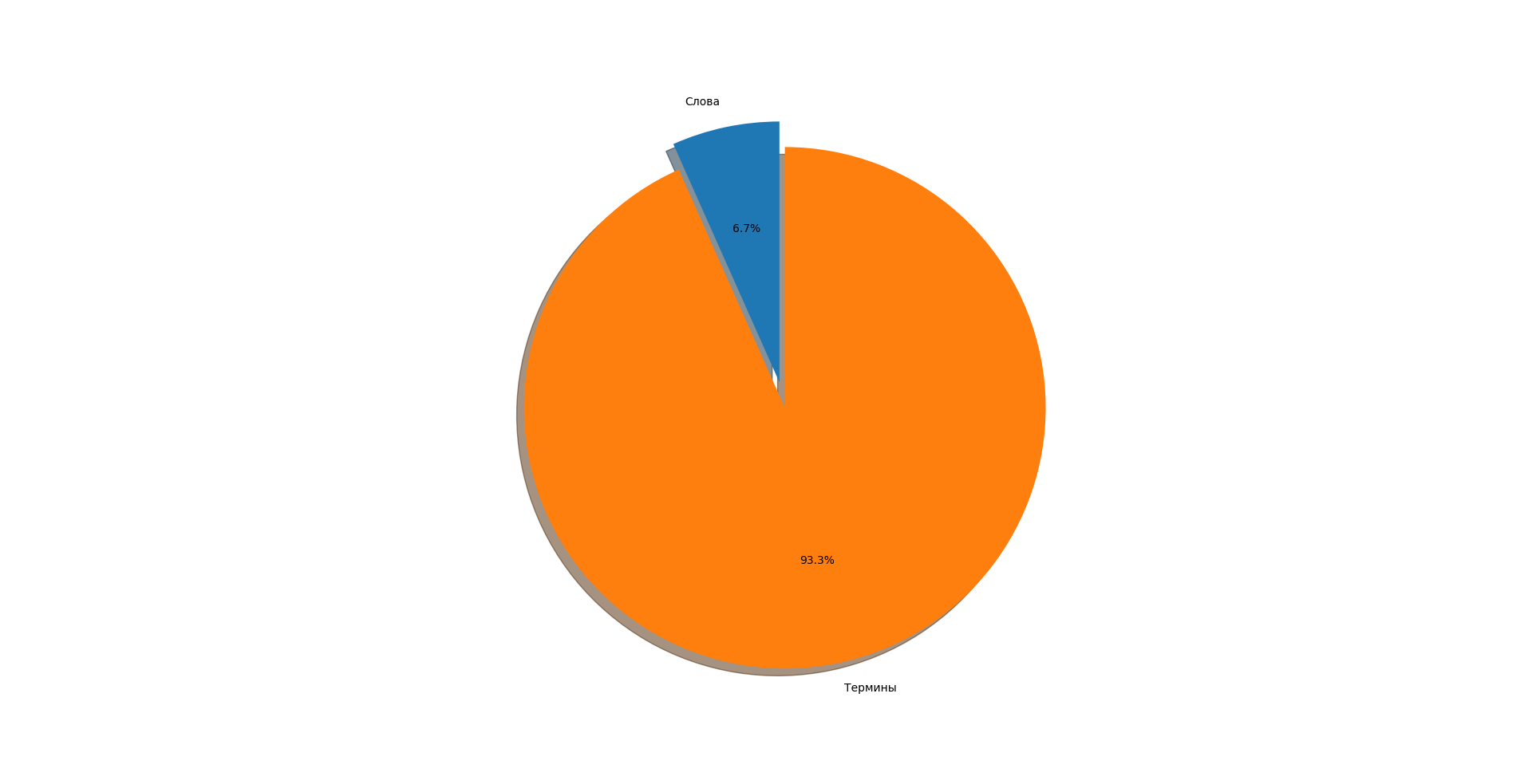
Хотя создание тезаурусов было основной визуализационной задачей в пределах этого корпуса, в процессе работы – часто просто для упрощения понимания получаемых результатов самим автором. В частности, во время работы над взвешенным извлечением терминов использовались такие диаграммы, которые выражали количество терминологии среди ключевых слов и их характер.



***Рис. 11. Диаграмма терминов «китайского» корпуса».***



***Рис. 12. Диаграмма терминов «медицинского» корпуса».***



***Рис. 12. Диаграмма ключевых слов «медицинского» корпуса». Оранжевая часть – термины, синяя – обычные слова.***

# Заключение

В ходе данной работы было проведено исследование разных сторон системы SketchEngine, были сделаны попытки улучшить и расширить ее функционал и справиться с имеющимися у нее проблемами. Хотя итоговый результат далек от совершенства, все же созданный инструментарий позволяет исследователю работать со SketchEngine так, как этого раньше не происходило, с учетом большего количества факторов и новыми, ранее не использованными способами.

Помимо практической пользы от собственно инструментария был также исследован процесс лемматизации языковых данных, которые в итоге приводят к созданию корпуса SketchEngine. Из собранных теоретических данных можно сделать вывод, что лемматизатор чаще всего "спотыкается" в двух случаях.

Во-первых, неправильно лемматизируются слова, которые не включаются в обычные словари, либо по причине их просторечности, либо из-за того, что они являются производными от более распространенных слов. Такие слова при этом могут встречаться весьма и весьма часто, из-за чего их неверная лемматизация и является большой проблемой. К системе можно подключать сторонние лемматизаторы, способные служить подспорьем для ограниченных возможностей самого SketchEngine.

Второй частой ошибкой для лемматизатора являются составные слова. Здесь решение проблемы не столь однозначно. Хотя расширение словаря может помочь с более устойчивыми составными словами ("диван-кровать"), все возможные составные слова учесть невозможно, так как сложение слов с помощью дефиса является продуктивным способом словообразования. Усугубляет проблему также то, что такие слова могут склоняться по-разному (иногда склоняются обе части, иногда - только последняя). Потенциальным решением может служить разделение составных слов надвое: в отдельности их части легче лемматизировать, при этом во многих случаях их семантическое значение примерно сохранится. "Диван-кровать" - примерно то же самое, что диван и кровать вместе, "Ростов-на-Дону" - действительно Ростов на Дону: во всяком случае, было бы вполне логично, если бы эти слова появлялись при поиске коллокаций по леммам "диван" и "дон".

Автор надеется, что эта работа продолжит развиваться и в будущем затронет и эти проблемы.

# 

# Литература

1. В.П. Захаров, С. Ю. Богданова, Корпусная лингвистика, СПб, 2013
2. Копотев М. Введение в корпусную лингвистику: Учебное пособие для студентов, Прага, 2014
3. Д. В. Сеченова. К задаче создания корпусов русского языка, Москва
4. Хохлова М.В., Исследование лексико-синтаксической сочетаемости в русском языке с помощью статистических методов, СПбГУ, Санкт-Петербург, 2010
5. Allan, Keith, the Oxford Handbook of the History of Linguistics (History of Corpus Linguistics), 2013
6. Allen, William. Making Corpus Data Visible: Visualising Text With Research Intermediaries, University of Oxford, Оксфорд, 2015
7. Barrs, Keith: Using the Sketch Engine Corpus Query Tool for Language Teaching : The Language Teacher - Issue 40.2; March 2016, Хиросима, Hiroshima Shudo University, 2016
8. Few S, Information Dashboard Design: The Effective Visual Communication of Data, Sebastopol, 2006
9. Gabrielatos, C. (2007). Representation of refugees and asylum seekers in UK newspapers, 1996-2005: Report on the corpus-based analysis. Joint meeting of the Corpus Research Group (CRG) and the Language Ideology and Power Research Group (LIP), Department of Linguistics and English Language, Lancaster University, 15 January 2007.
10. Gatto, Maristella, Web As Corpus: Theory and Practice, Bloomsbury Academic, London, 2014
11. Golik, Bossy, Ratkovic and Nédellec, Improving term extraction with linguistic analysis in the biomedical domain
12. Kantner, Cathleen; Kutter, Amelie, How to get rid of the Noise in the Corpus: Cleaning Large Samples of Digital Newspaper Texts, Stuttgart, 2011
13. Kilgarriff, Marcowitz, Smith, Thomas: Corpora and Language Learning with the Sketch Engine and SKELL
14. Kilgariff, Adam; Baroni, Marco, WebBootCaT: a web tool for instant corpora, Brighton, 2006
15. Adam Kilgarriff, VítBaisa, Jan Bušta, MilošJakubíček, VojtěchKovvář, Jan Michelfeit, PavelRychlý, VítSuchomel. TheSketchEngine: tenyearson. Lexicography, 1: 7-36, 2014.
16. Adam Kilgarriff, PavelRychlý, PavelSmrž, David Tugwell. Itri-04-08 the sketch engine. Information Technology, 2004.
17. Kilgarriff, Finding Terms in Corpora for Many Languages with the Sketch Engine, Lexical Computing Ltd., United Kingdom
18. Meunier F., Gilquin Gaëtanelle, Paquot Magali, A Taste for Corpora: In Honour of Sylviane Granger, John Benjamins Publishing Company
19. Nesselhauf, Nadja, Corpus Linguistics: A Practical Introduction, Гейдельберг, 2005
20. Talianová, Dominika. Visualization of Corpus Data, Брно, 2014
21. Sophie Aubin, Thierry Hamon. Improving Term Extraction with Terminological Resources. Tapio Salakoski, Filip Ginter, Sampo Pyysalo, Tapio Pahikkala. 2006, Springer, pp.380, 2006, LNAI 4139. <hal-00091444>
22. Statistics used in Sketch Engine, <https://www.sketchengine.co.uk/documentation/statistics-used-in-sketch-engine/>
23. Zakharov V.P, Automatic Collocation Extraction: Association Measures Evaluation and Integration, Saint-Petersburg State University, Saint-Petersburg, Russia
24. Computational Linguistics and Intellectual Technologies, Papers from the Annual International Conference, “Dialogue” (2010) Issue 9 (16)