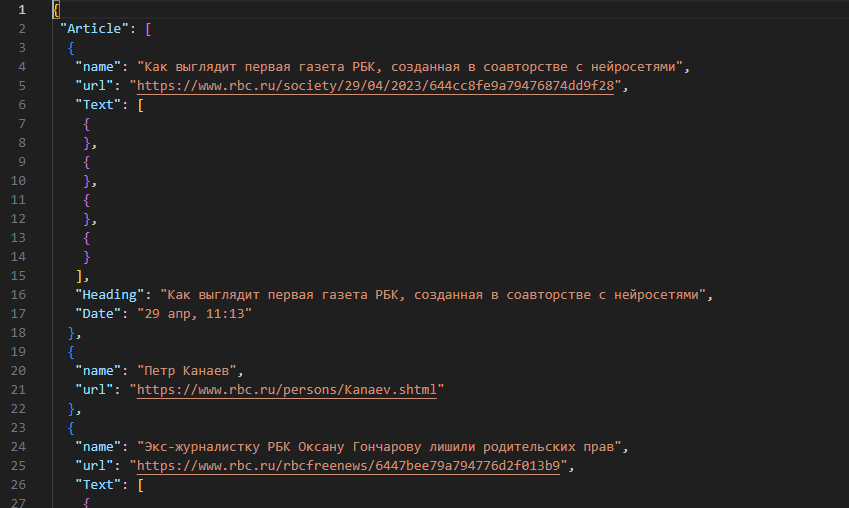
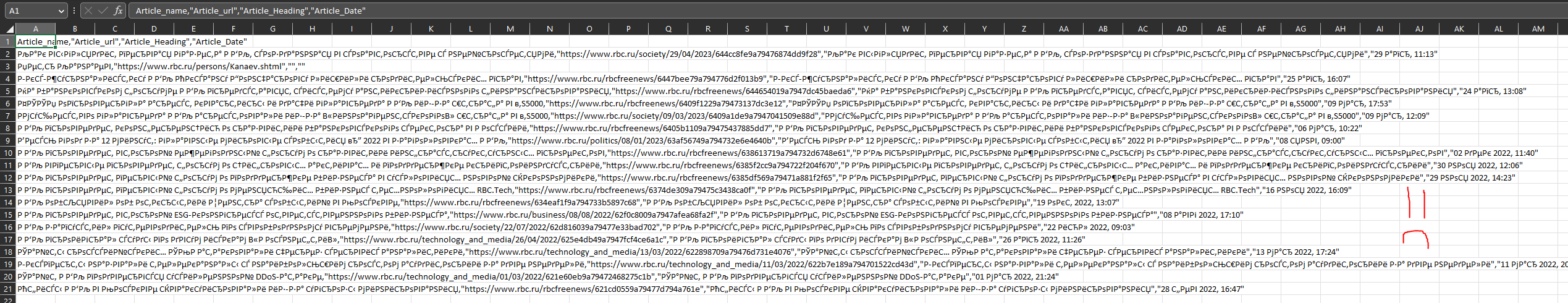
Уровень 1

Лабораторная работа 1

В первой лабораторной мы должны были познакомиться с парсингом, а если конкретнее, то с ресурсом ParseHub. После несложной регистрации меня встречали многочисленные туториалы, которые я прошёл. Дальше к самому заданию. Выбрал я новостной сайт РБК (<https://www.rbc.ru/tags/>), так как он, как нельзя лучше подходил по заданию и хотел получить информацию с него. Проделав тот же алгоритм, что и в задании (далеко не с первой попытки) я получил выгрузку данных. Если в эксель всё непонятно, то на JavaScript (открытом через VSC) всё было куда попроще, поэтому прикрепляю два файла, возможно данных, представленных там недостаточно, но это – всё, что он мне выдал, хотя сама страница РБК обновлялась вниз до бесконечности (может какое ограничение, а может ей сложно воспринимать в таком формате, не знаю). В любом случае, отмечу, что инструкция понятная, да и само приложение не сказать, что сложное и понять можно всё быстро. Результаты теста приложил в файлах с названием 1run\_results (.csv и .json)





Лабораторная работа 2

Далее поработаем с программой MyStem. Скачаем сам .exe файл, закинем его по пути C:\Stem и создадим там же input.exe и output.exe. Закинем в input.txt, например, произведение Фонвизина “Выбор гувернера” целиком (файл “2Фонвизин.txt”). В консоль закинем сначала путь, прописав cd C:\Stem. Сначала пролемматизируем и удостоверимся в работе, прописав **mystem.exe input.txt output.txt**. получим данные, представленные в файле “2лемматизированные данные.txt”. Далее самое главное – разметим текст. Сделаем это с помощью команды mystem.exe input.txt output.txt -i -n -d. Получим данные, представленные в файле “2размеченные данные”. Далее нам нужна статистика. Путём несложных действий получим:

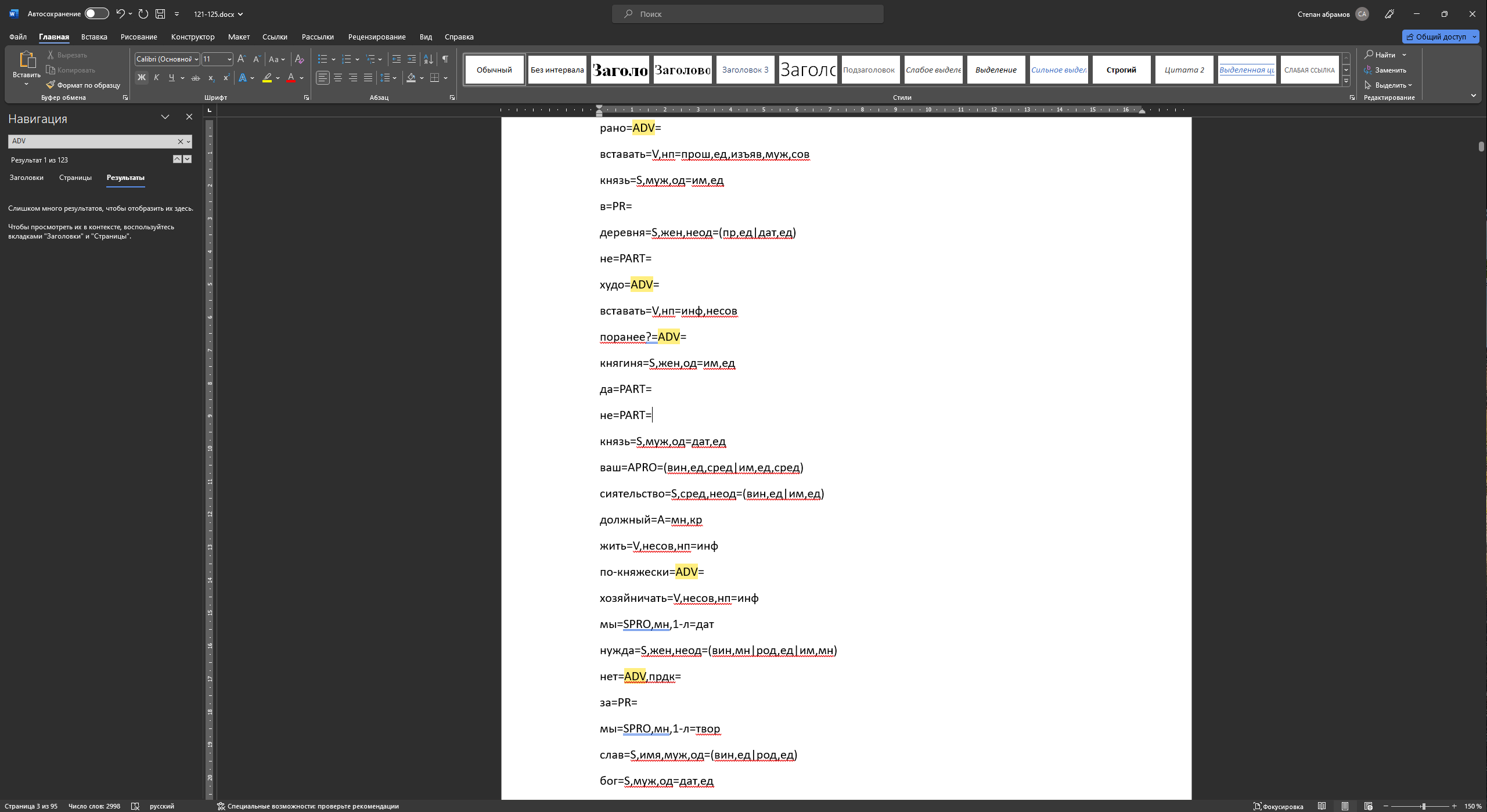
Местоимений (SPRO) – 373

Существительных (S) – 893

Прилагательных (A) – 707

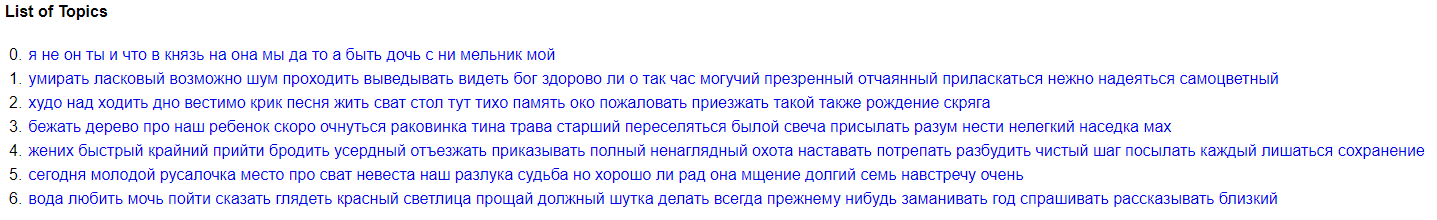
Глаголов (V) – 686

Наречий (ADV) – 123

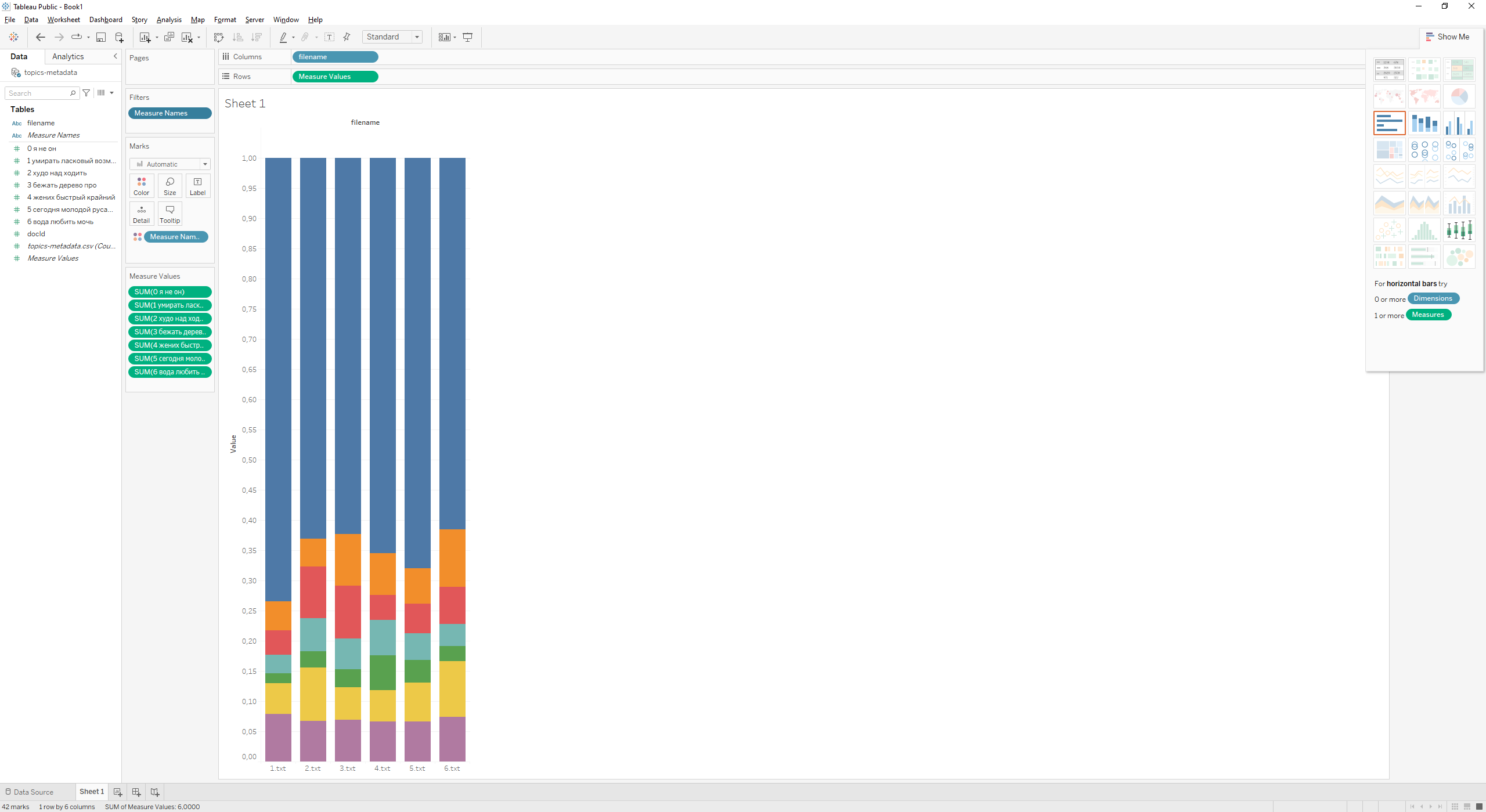


Лабораторная работа 4

Далее – тематическое моделирование. Возьмём за основу произведение Пушкина “Русалка” (файл “ 4Русалка.txt”). С помощью функций -w -l -d пролемматизируем данные в MyStem, предварительно, разделив текст на 6 глав (“1-input.txt”, “2-input.txt” и т.д.). Далее уберём фигурные скобки и перенесём всё в файлы “1-output.txt”, “2-output.txt” и т.д. После запуска программы TopicModelingTool, делая всё по инструкции, я действительно получил нужные выводы (в папке “TMT output”):

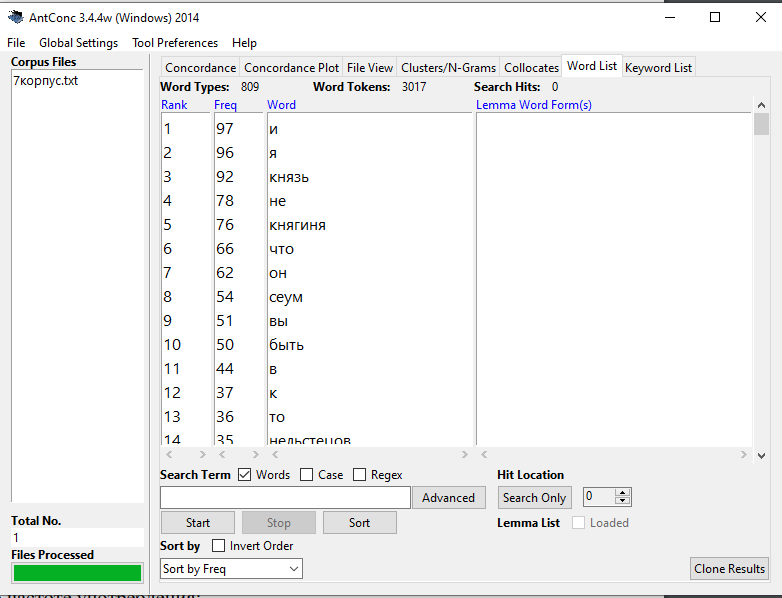


Лабораторная работа 11

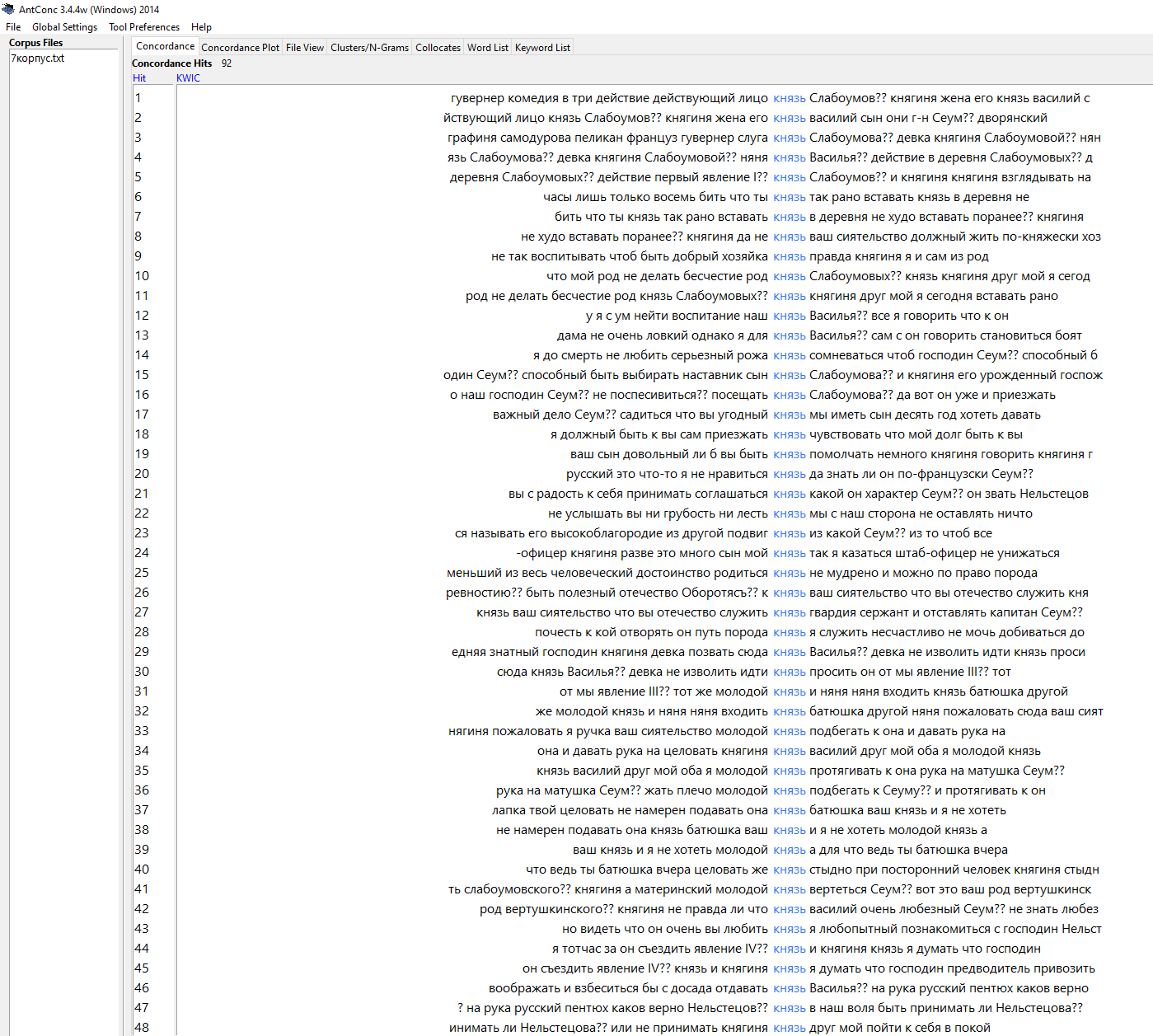
После поработаем с Tableau. Откроем наши топики в формате .csv. Далее, в соответствии с мануалом, получим следующее графическое представление, в соответствии с 6 главами (они же 6 файлов) я получил 6 столбцов:  


Уровень 2

Лабораторная работа 7

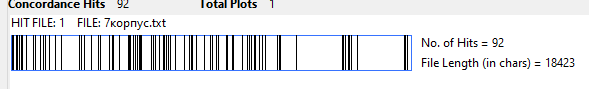
Возьмём за наш корпус уже знакомый нам текст Фонвизина. В AntConc очень быстро получим список самых используемых словоформ:  


Далее найдём употребления самого частотного слова (не стоп-слово, а то с ним не интересно, то есть слово князь):

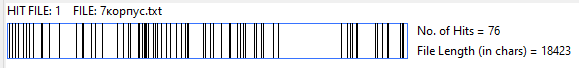


Далее – позиционные графики слов. В большинстве случаев также будем работать не со стоп-словами, но пару раз их возьмём сравнить результаты.

Слово “князь”:



Слово “княгиня”:



Слово “сеум” (неизвестно, что это и почему так часто встречается):



Далее для разнообразия пара стоп-слов. Слово “и”:



Слово “я”:



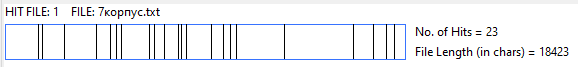
Слово “быть”:



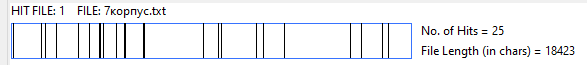
Слово “нельстецов”:



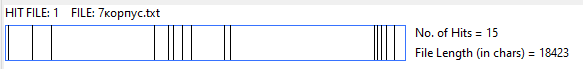
Далее слова, которые не вошли в десятку и их не видно на скриншоте. Слово “господин”:



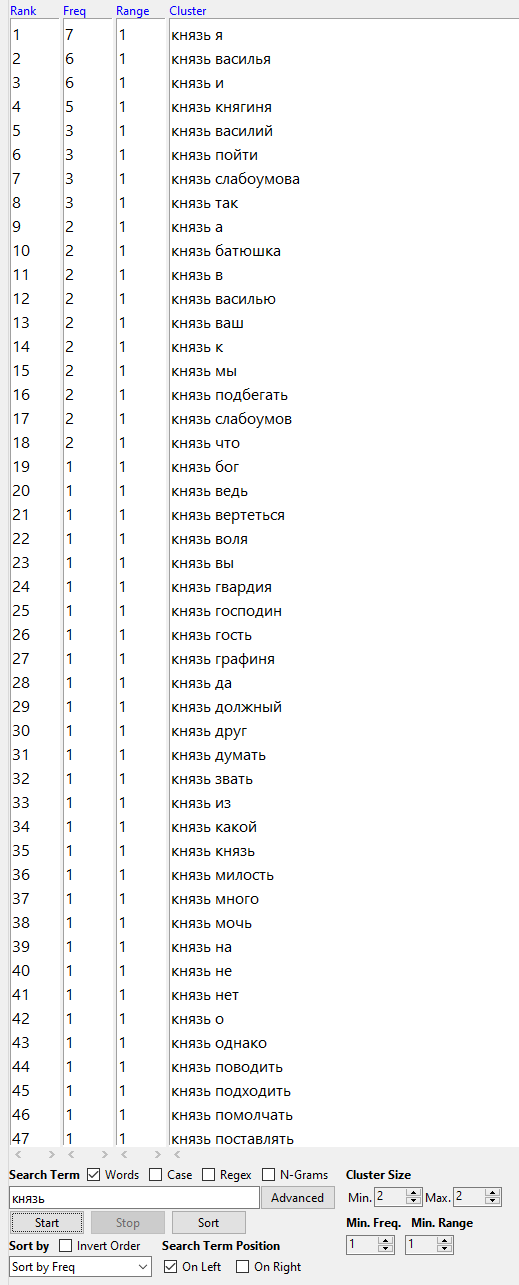
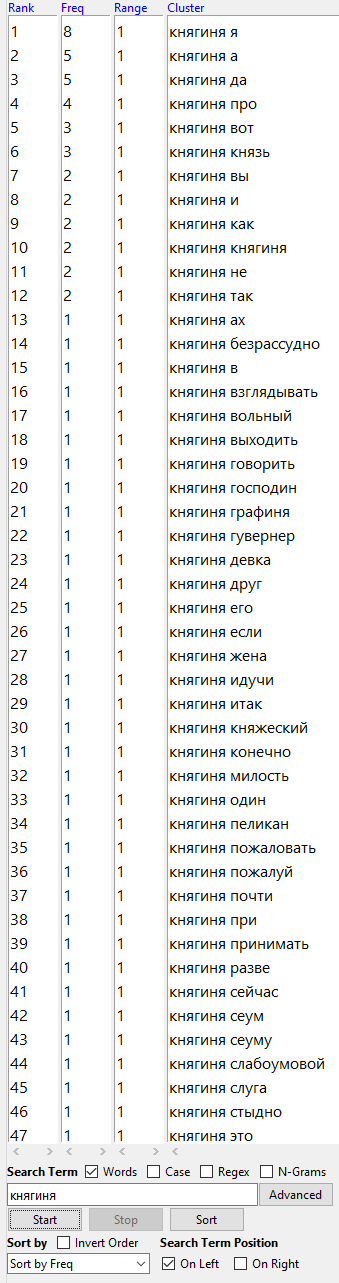
Слово “сын”:



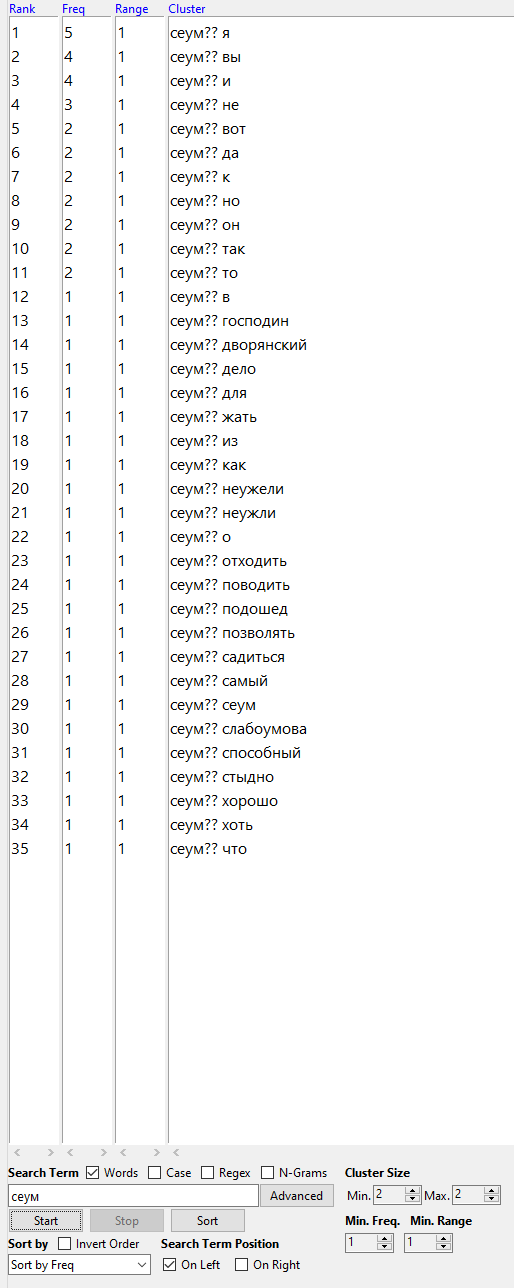
И финальное слово – “предводитель”:



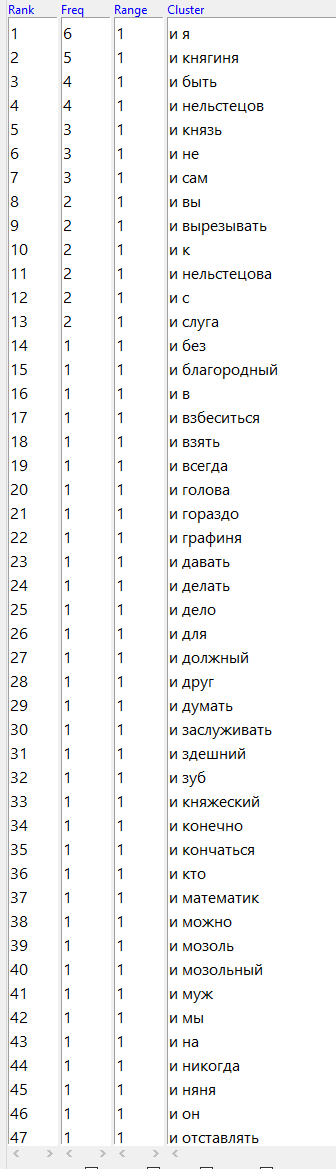
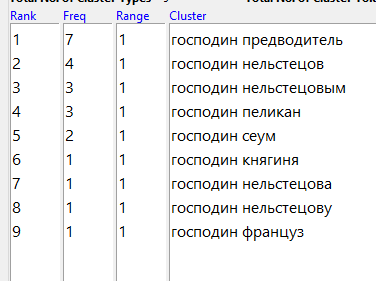
Далее создадим список кластеров:

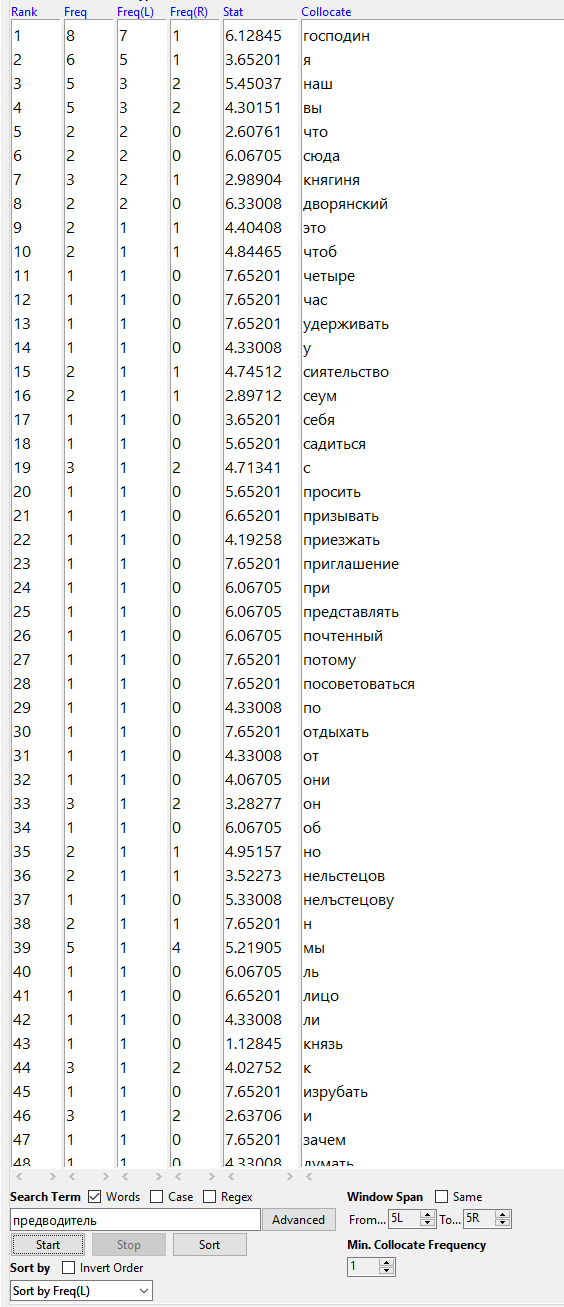
Этот загадочный сеум, который после кластеризации так загадочным и остался:



И ещё два слово, включая стоп-слово “и”:

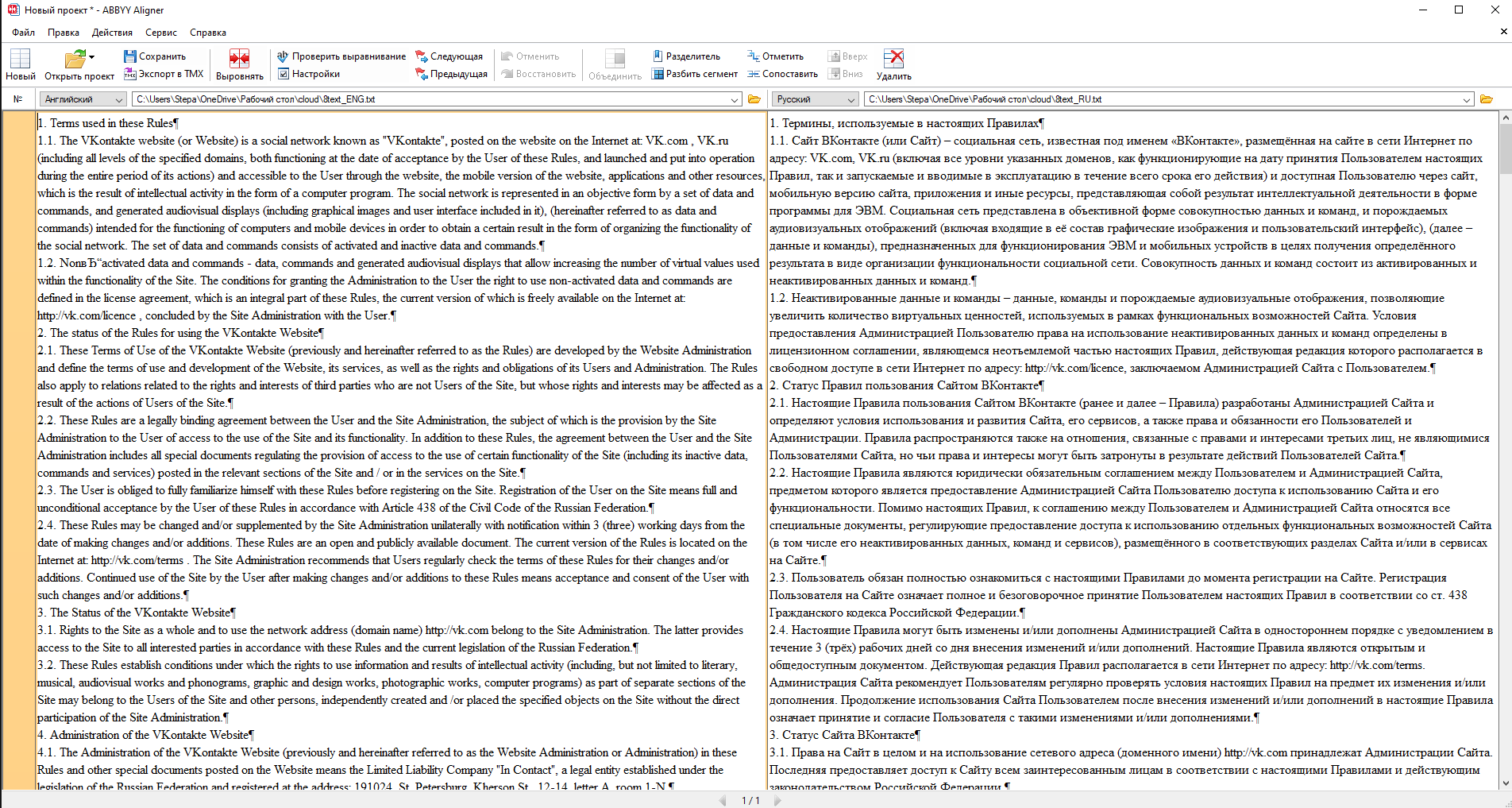
 

И, наконец, коллокации со словом “предводитель”:

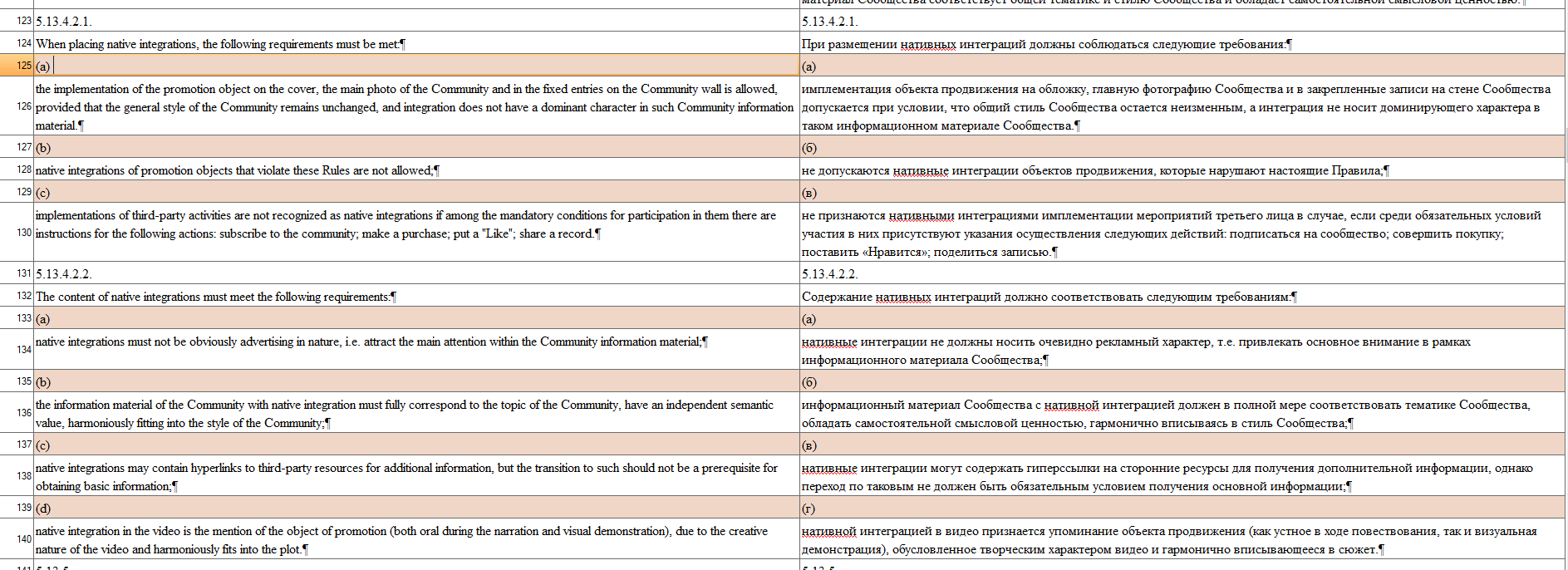


Лабораторная работа 8

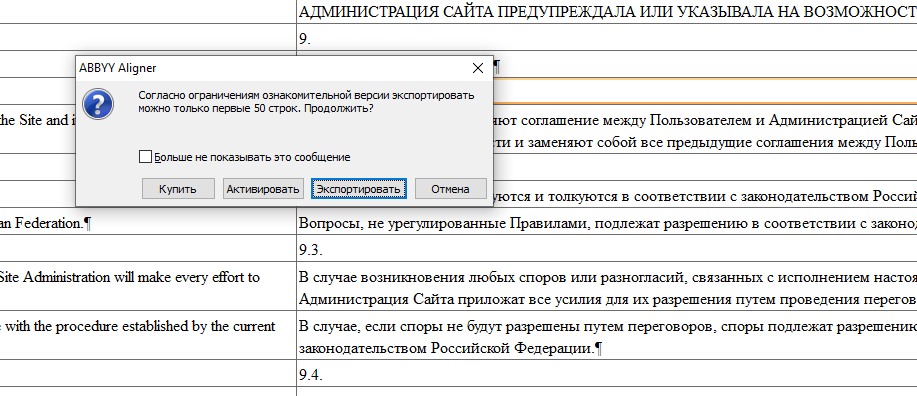
Для работы в ABBYY Aligner нам понадобится объёмный нехудожественный текст. Я взял пользовательское соглашение ВК объёмом 6072 слова и перевод на 7732 (8text\_RU.txt и 8text\_ENG.txt соответственно). Тексты в программе:



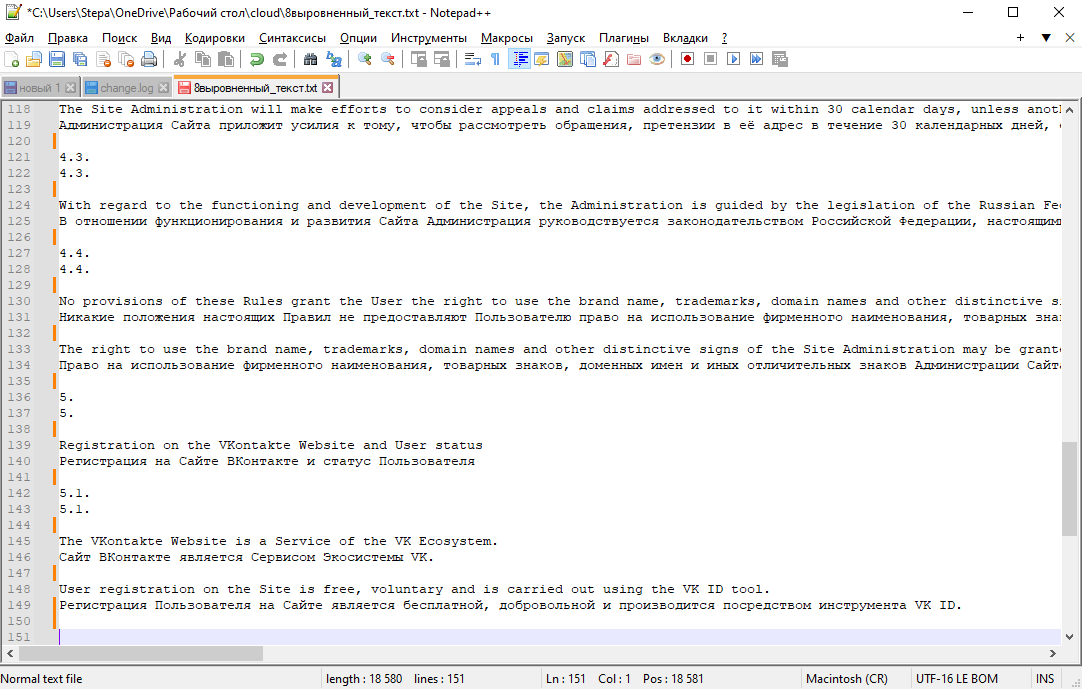
Результаты выравнивания:



Неуверенной программа была почти только в подпунктах соглашения, то есть в отдельных буквах, например, (а). Поможем же программе. Сливать же строки почти не понадобилось. После, устранив все неполадки и ошибки, сохраним файл в формате .rtf (8text\_ENG.rtf). К сожалению, бесплатная версия предлагает сохранение лишь первых 50 строк:

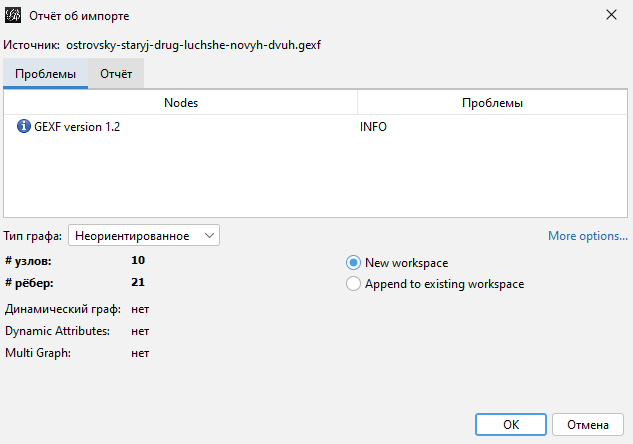


Но я также сохранил полную версию, но в формате .aga (8text\_ENG.ata).

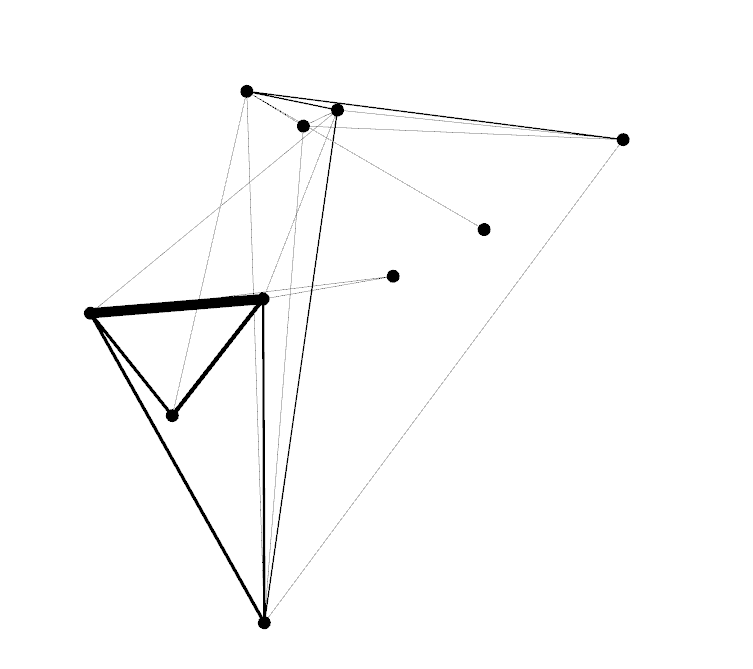
Далее конвертируем .RTF файл в .TXT и, воспользовавшись программой NotePad++ удалим пустые строки и запишем макрос для добавления пробела между каждой парой оригинал-перевод. Вот так будет выглядеть текст с применённым макросом:  


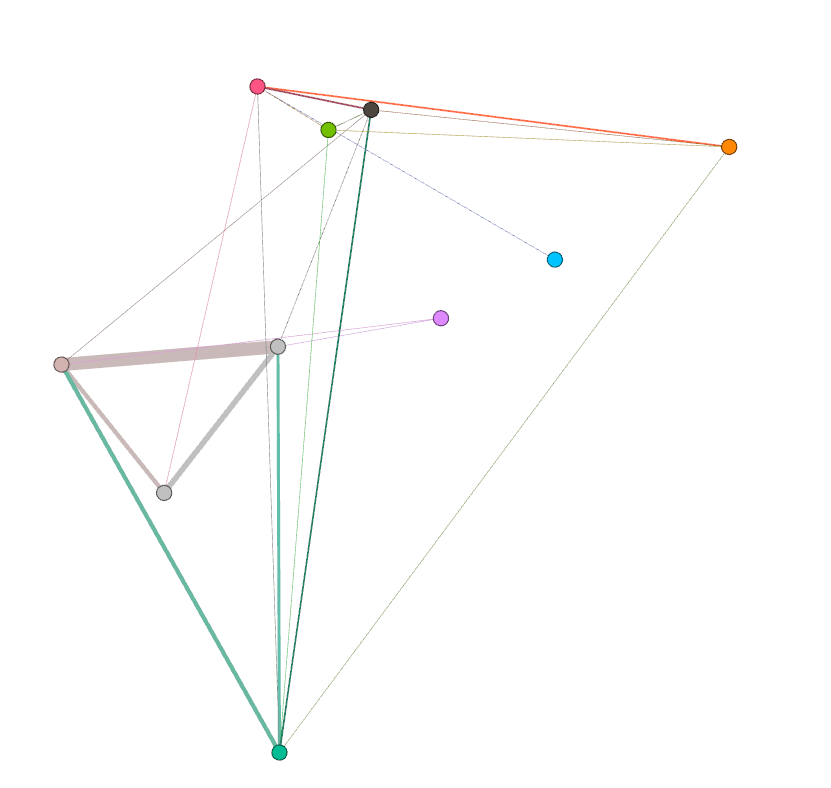
Лабораторная работа 10

Далее – построение графов в Gephi. Возьмём в этот раз произведение Островского “Старый друг лучше новых двух” (ostrovsky-staryj-drug-luchshe-novyh-dvuh.gexf) и засунем его в Gephi. Получим 10 узлов и 21 ребро:

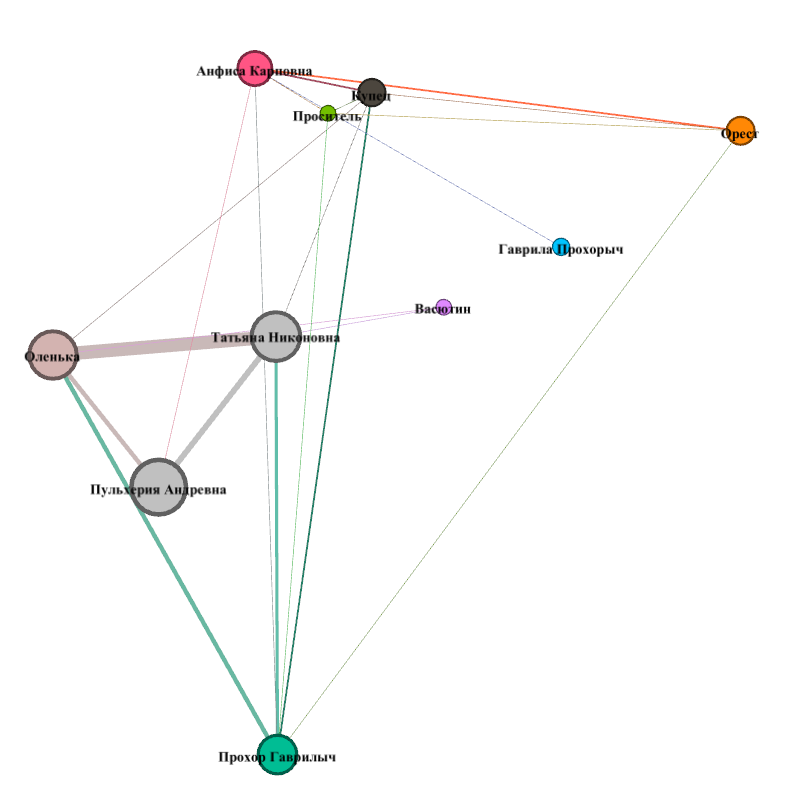


И сам граф:

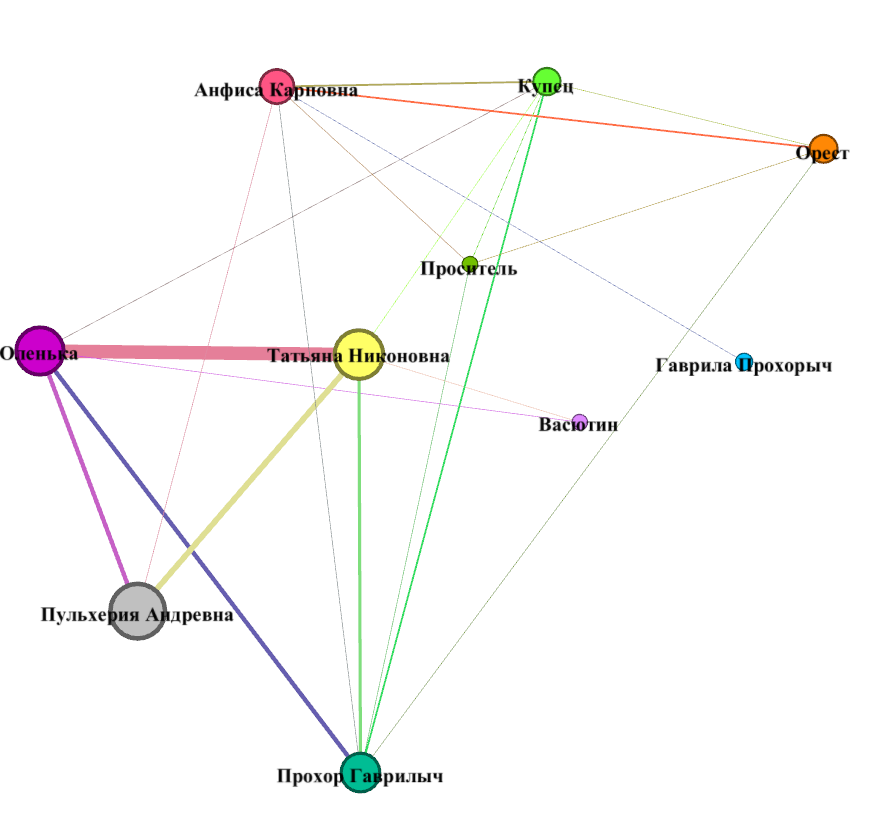


Изменим цвет рёбер в соответствии с суммарным количеством слов во всех репликах:  


Далее включим отображение подписи узлов, настроим шрифт и размер узлов:



И финальный вид графа:



Ресурсы Pullenti

SDK Pullenti предназначено для разработчиков информационных систем, работающих с текстами на естественном языке. SDK представлено в исходных кодах на языках программирования C#, Java, Python и JavaScript.

SDK состоит из трех независимых частей: Lingvo, Unitext и Address. Lingvo предоставляет лингвистический анализ текстов, включая морфологический анализ, выделение именованных сущностей, семантический анализ и другие функции. Address занимается выделением и привязкой адресов к объектам ГАР ФИАС. Unitext отвечает за выделение текстовых данных и структурирующих элементов из файлов различных форматов.

Алгоритмы, используемые в SDK Pullenti, основаны на правилах, без применения машинного обучения. SDK Address работает с локальным индексом, основанным на данных ГАР ФИАС. SDK Unitext поддерживает множество форматов файлов, включая docx, doc, pdf, rtf, odt, html и другие.

SDK Pullenti предоставляет разработчикам возможность работать с текстовыми данными, выполнять лингвистический анализ, извлекать адреса и структурировать текстовую информацию из различных файлов.

RapidMiner – это платформа для анализа данных и машинного обучения, которая предоставляет широкий набор инструментов и функций для разработки и выполнения аналитических процессов. Он предназначен для специалистов по анализу данных, ученых и разработчиков, которые работают с большими объемами данных и стремятся извлечь ценные познания и модели из своих данных.

Вот некоторые общие возможности, которые предоставляются RapidMiner:

Широкий набор операций для предварительной обработки данных, таких как удаление дубликатов, заполнение пропущенных значений, нормализация, фильтрация и многое другое.

Инструменты для визуализации данных, которые позволяют вам исследовать структуру и распределение ваших данных.

Спектр алгоритмов машинного обучения, включая классификацию, регрессию, кластеризацию, ассоциативные правила и другие.

RapidMiner позволяет вам создавать и автоматизировать аналитические процессы.