Санкт-Петербургский государственный университет

ГАВРИЛИК Дарья Александровна

Выпускная квалификационная работа

ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ МЕТОДОВ АВТОМАТИЧЕСКОГО ВЫДЕЛЕНИЯ КЛЮЧЕВЫХ ВЫРАЖЕНИЙ В КОРПУСАХ РУССКОЯЗЫЧНЫХ ТЕКСТОВ

Уровень образования: магистратура

Направление 45.04.02 «Лингвистика»

Основная образовательная программа ВМ.5805.

«Компьютерная и прикладная лингвистика»

Профиль «Компьютерная лингвистика»

Научный руководитель:
доцент, Кафедра математической лингвистики,
Митрофанова Ольга Александровна

Рецензент:
доцент, ФГБОУ ВО
"Сыктывкарский
государственный
университет имени
Питирима Сорокина",
Хозяинов Сергей Александрович

Санкт-Петербург
2021

Оглавление

Введение

Увеличение потока текстовой информации в современном мире порождает необходимость эту информацию структурировать, упорядочивать, делать более простой и быстрой для поиска. Именно для решения задач компрессии потока текстовой информации необходимы методы автоматической обработки текстов. Одной из важнейших таких задач является процедура автоматического извлечения ключевых выражений из текстов для рубрикации, индексировании, классификации, бизнес-стратегиях, лексикографии, библиотечном деле, информационном поиске и так далее. Ключевые выражения также помогают составить быструю оценку содержания документов, что становится актуальнее в настоящем мире.

Исследования в области автоматического извлечения ключевых выражений широко проводятся на материале английского языка. Существует множество алгоритмов и исследований особенно в последние десятилетия. Исследования с использованием русскоязычного материала можно найти у таких исследователей, как Е.В.Ягунова, О.А.Митрофанова, Т.Ю.Шерстинова, А.Д.Москвина и др. Довольно часто предпринимаются попытки адаптирования алгоритмов для работы с англоязычными документами под работу с русскоязычными. Однако сопоставительных исследований работы разных алгоритмов не было проведено и оценено на материале русского языка. В настоящем исследовании впервые описываются алгоритмы, способные работать с русскоязычными документами смешанных функциональных стилей, с дальнейшей обработкой результатов оценки эффективности.

Актуальность работы обуславливается необходимостью структурирования потока текстовой информации при помощи компрессии. В данной работе проводится исследование природы текстового документа на предмет местоположения ключевого выражения относительно традиционного деления текста на введение, основную часть и заключение. Таким образом,

деление документа на части и извлечение ключевых выражений из определенной его части в разы сократит время обработки текста на естественном языке.

Извлечение ключевых выражений из документов является довольно традиционной, а значит, старой процедурой, но с изменением типа текстов и задач работы с этим текстом, актуальным является сравнение методов с целью определения их эффективности.

Новизна исследования заключается в использовании сопоставительного анализа на смешанных корпусах разных функциональных стилей – публицистическом, научном и художественном. Впервые проводится исследование сравнения не только самих алгоритмов на предмет эффективности, но и сравнение ключевых выражений, извлеченных алгоритмами и размеченных экспертами. Данное решение обусловлено целью исследовать природу текстов и определения местоположения ключевых выражений в тексте.

Объектом исследования выступает природа ключевых выражений в корпусе русскоязычных текстов смешанных стилей. **Предметом** настоящей работы являются методы, использованные для извлечения ключевых выражений из текстов.

Цель исследования состоит в том, чтобы экспериментальным путем определить местоположение ключевых выражений относительно всего текста при помощи сравнения экспертной разметки и различных методов автоматического выделения ключевых выражений при работе с русскоязычными текстами различной тематики и стилей.

Для достижения данной цели требуется решить следующие **задачи**:

1. исследовать теоретические основания процедуры автоматического выделения ключевых выражений, проанализировать подходы к выделению ключевых выражений с точки зрения психолингвистики (А.С.Штерн, Л.В.Сахарный, Л.Н.Мурзин, Е.В.Ягунова и др.);
2. произвести отбор методов автоматического выделения ключевых выражений для проведения экспериментов (tf-idf, Log-likelihood, Chi-

square, RAKE, YAKE, TextRank, KeyBERT, Topia, PullEnti), дать характеристику каждого из методов и обосновать свой выбор;

3. подготовить лингвистические данные для проведения экспериментов: произвести сборку и предобработку исследовательских корпусов текстов разных стилей (художественный, научный, публицистический);
4. разработать процедуру сравнения разметки экспертов и методов автоматического выделения ключевых выражений;
5. произвести планирование и проведение экспериментов:
    a) определить параметры экспериментов: определить объемы текстов, длину ключевых выражений, объемы списков ключевых выражений, способы их ранжирования и т.д.
    b) автоматически извлечь ключевые выражения из корпусов текстов;
    c) извлечь ключевые выражения при помощи экспертов;
    d) разработать и провести процедуры оценки результатов;
    e) сравнить данные, полученные с помощью исследуемых методов автоматического выделения ключевых выражений и эталона.
6. проанализировать результаты экспериментов: определить местоположение ключевого выражения относительно всего текста.

Материалом исследования является русскоязычный корпус, состоящий из трех подкорпусов разных функциональных стилей: публицистического, научного и художественного. В каждом подкорпусе содержится 50 документов, то есть в сумме корпус составляет 150 текстов на русском языке. Сборка корпуса производилась автоматически и вручную, предварительная обработка текста проводилась при помощи графематического анализа с удалением таблиц, рисунков и так далее. Особенностью каждого документа в подкорпусах является наличие аннотации (развернутый заголовок для новостного текста, собственно аннотация и ключевые слова для научной статьи и опорные слова, вынесенные автором произведения перед каждой главой с сюжетными событиями повествования).

В настоящем исследовании была выдвинута гипотеза о существовании зависимости ключевого выражения и его местоположения. В данном случае проверялось наличие ключевых выражений в начале текста.

Для проверки или опровержения гипотезы был проведен следующий эксперимент:

1. тексты подкорпусов функциональных стилей разбиты на 2 части – начало и остаток. Для текста каждого функционального стиля было собственное деление ввиду разной природы текстов. Для публицистического подкорпуса - заголовок и первые два-три предложения новости, для научного - аннотация и первый абзац статьи, для художественного - размеченные автором опорные слова и первый абзац главы;
2. первые части текстов размечены экспертами;
3. вторые части текста автоматически обработаны автоматическими методами извлечения ключевых выражений;
4. произведена процедура сравнения ключевых выражений, размеченных экспертами и извлеченных алгоритмами. Таким образом, проверялось количество совпадений результатов алгоритма с результатами эталона – экспертной разметкой;
5. проведена процедура оценки эффективности по каждому методу.

Теоретическая значимость исследования заключается в обосновании существования зависимости местоположения ключевых выражений относительно всего текста. С одной стороны, исследуется природа и структура построения текста. С другой стороны, исследуется эффективность извлечения ключевых выражений алгоритмами, способными работать с русскоязычными документами.

Практическая значимость исследования заключается в реализации собственного экстрактора ключевых выражений, основанного на грамматических правилах. В рамках настоящей работы на материале научного подкорпуса было проведено исследование структуры ключевых выражений.

Выяснилось, что у ключевого выражения научного подкорпуса есть ограничения в виде отсутствия ключевых выражений, состоящих из глагольной группы.

Объем и структура диссертации. Работа состоит из 7 глав, введения, заключения, списка источников и 4 приложений. Главы 1 и 2 посвящены теоретическим вопросам: проблемам, возникающим при извлечении ключевых выражений, а также обзору научной литературы по избранным методам автоматического извлечения ключевых выражений. Глава 3 содержит информацию о лингвистических данных, необходимых для проведения эксперимента, то есть о сборке корпуса и структуре документов. В главе 4 приводится методика проведения экспериментов с описанием параметров, необходимых для экспериментов. Глава 5 посвящена непосредственно проведению самого эксперимента с разбиением на параграфы подкорпусов разных функциональных стилей. В главе приводятся примеры текстов, ключевые выражения экспертов и методов, а также результаты совпадений разметки информантов и выдачи алгоритмов. В главе 6 содержатся результаты оценки эффективности полученных результатов совпадений. Завершающая 7 глава посвящена исследованию природы ключевого выражения на материале научного подкорпуса, а также описанию разработанного самостоятельно экстрактора ключевых выражений. Приложения содержат листинги программ, использованных для проведения исследования на разных этапах работы. Общий объем работы составляет 143 страницы, основное содержание изложено на 120 страницах, текст содержит 44 таблицы, 5 рисунков и 99 формул, приложение занимает 18 страниц. Список источников состоит из 31 позиции.

Глава 1. Теоретические основания процедуры автоматического извлечения ключевых выражений

Ключевые выражения помогают определить уникальность любого письменного текста, рассказать читателю еще на этапе отбора информации, каково содержание того или иного текста. Ключевые выражения определяют тему, которая рассматривается в тексте, а также помогают читателю составить общее представление, так сказать информационный портрет документа. В работе Е.В. Ягуновой утверждается, что «извлечение наиболее важной информации, передаваемой текстом, ***может*** быть смоделировано через процедуры выделения ключевых слов (КС) текста.» [11]

Отдельной областью знаний в компьютерной лингвистике является процесс автоматического извлечения ключевых выражений, целью которого является список ключевых выражений, полученный в результате работы алгоритма. Данная процедура в разы сокращает время, затраченное человеком, а также помогает во многих областях, например, лексикографии, библиотечном деле, терминоведении и особенно информационном поиске. Эффективных результатов автоматического выделения ключевых выражений можно добиться в задачах обработки документов: индексировании, реферировании, классификации. Актуальность рассматриваемой нами задачи порождает большое количество исследований, уже проведенных и проводимых в настоящий момент, однако, автоматическое извлечение ключевых выражений представляет собой нерешенную проблему. Особенно проблематичным является автоматическое извлечение многокомпонентных ключевых выражений [10]. В частности, извлечение таких лексических групп, как именные группы, будет являться вызовом для исследователя данной области.

Процесс выделения ключевых выражений являет собой глубокий анализ текста, как для человека, так и для алгоритма. Эта задача реализуется по-разному, однако результатом всегда будет являться ограниченное количество выражений из данного документа, наиболее четко, полно и ясно

характеризующих этот документ. О том, как справляется с этой задачей человек, можно сказать одно – он действует более абстрактно и склонен к обобщению информации, используя синонимы и другие слова, не использующиеся в тексте. Что касается автоматической процедуры, здесь важен четкий алгоритмический подход для поиска тех самых малейших частиц текста, по праву называющихся ключевыми выражениями. Именно поэтому процесс извлечения ключевых выражений должен быть выполнен систематически и желательно без участия человека.

Автоматическое извлечение ключевых выражений включает в себя много разных процессов, происходящих на отдельных этапах всего алгоритма. В общем виде само извлечение выглядит следующим образом: отбор текста (текстов), предобработка текста (текстов), состоящая из удаления стоп-слов и лемматизации, непосредственно работа алгоритма. Стоп-словами являются слова, не несущие никакой смысловой нагрузки, то есть артикли, предлоги, союзы, частицы, местоимения, вводные слова, междометия и т.д.).

При автоматическом извлечении ключевых выражений могут использоваться источники дополнительной информации, например, тематический корпус, обучающая выборка с разметкой ключевых выражений, WordNet, онтологии и т.д. Результатом выдачи могут быть ключевые выражения-униграммы (при использовании фильтров для терминов, для извлечения бытовых понятий и т.д.), ключевые выражения-биграммы/триграммы и т.д., а также смешанная выдача.

Кандидаты ключевых выражений отбираются в виде n-грамм, не разделенных знаками препинания (кроме дефиса и кавычек) и стоп-словами, где n-грамма – это термин из компьютерной лингвистики, означающий последовательность из n элементов текста, например, слово или их последовательность.

Для каждого из кандидатов ключевых выражений рассчитываются признаки, которые позволяют судить о важности кандидата в данном документе. Набор кандидатов ключевых выражений з ранжируется по значениям признаков,

например, в соответствии с их частотностью и весами информативности, рассчитанными по одной из методик или с использованием определённого подхода.

После ранжирования производится отбор первых лучших ключевых выражений из этого списка или отбираются кандидаты, превышающие установленный минимальный порог значения признака.

Расчет веса информативности в данном случае играет важнейшую роль, поскольку этот показатель позволяет оценить значимость ключевого выражения по отношению к другим выражениям в документе.

Существует множество подходов, использующихся для автоматического извлечения ключевых выражений. Одними из самых распространённых являются статистические и лингвистические подходы, подходы, основанные на машинном обучении и т.д. В отдельный алгоритм может входить сочетание разных методов из перечисленных подходов, что помогает алгоритму глубже анализировать текст и выдавать более точный результат. В случае с ключевыми выражениями точность является более информативным и важным показателем, поскольку показывает, сколько автоматически выделенных ключевых выражений находятся в эталонной выборке.

Из перечисленных подходов вытекают определённые методы, которые извлекают ключевые выражения согласно определенным алгоритмам и идеям авторов того или иного алгоритма. Существует большое количество методов, активно использующихся для решения настоящей задачи. На рисунке 1 приведена схема четырех подходов и основанных на них методов. Это далеко не конечный список, однако, здесь представлены наиболее встречающиеся в научной литературе методы.

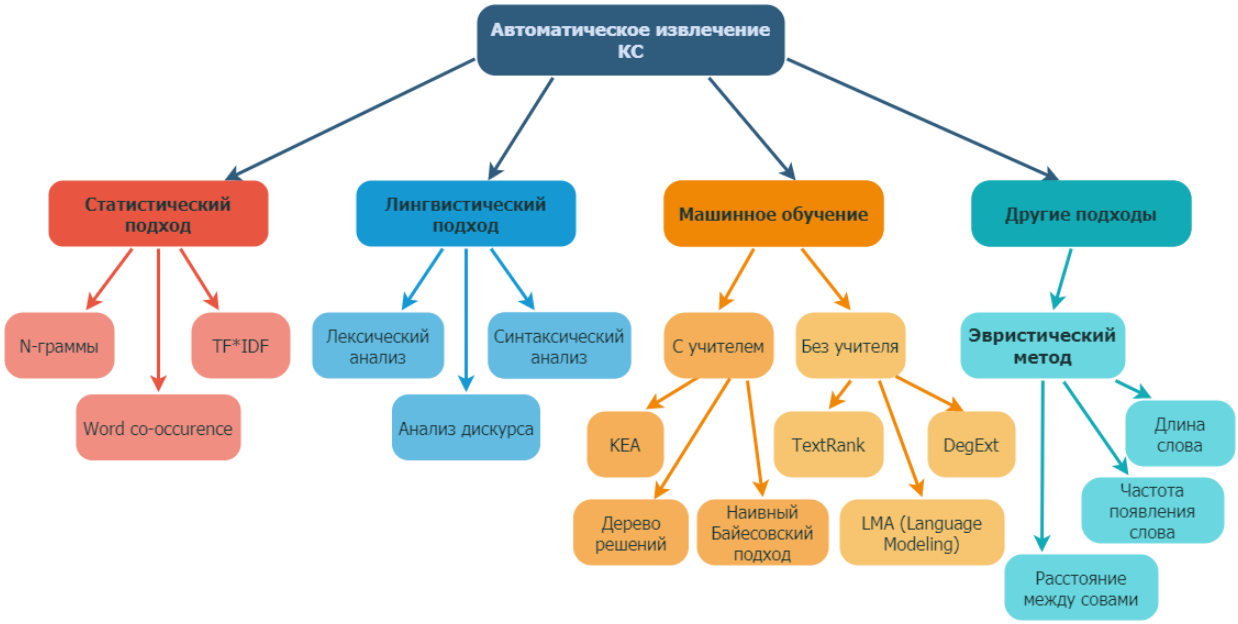

Рисунок 1. Схема подходов и некоторых методов автоматического извлечения ключевых выражений

1.1. Проблемы, возникающие при извлечении ключевых выражений

Процедура извлечения ключевых выражений является зависимой от преследуемой задачи. Ключевые выражения для задач информационного поиска будут кардинально отличаться от ключевых выражений, которые необходимы ученому для публикации научной статьи. А также важным остается вопрос о природе частей речи, которые могут быть избраны в качестве кандидатов на ключевые выражения, а затем приведены в списке ключевых выражений с самыми большими весами. То есть могут ли глаголы или глагольные группы считаться ключевыми выражениями для какой-либо задачи или главенствующее место займут имена существительные и именные группы?

Таким образом, для решения задачи извлечения ключевых выражений из текстов необходимо найти ответы на следующий ряд вопросов.

1. Какова цель извлечения ключевых выражений, то есть для чего конкретно нужны слова?
2. Тексты какого функционального стиля будут использованы в качестве объектов?

3. Чем является ключевое выражение при решении настоящей задачи, то есть необходимо формально определить части речи?
4. Каковы детали выдачи ключевых выражений (количество, ранжирование и т.д.)?

В работе [15] приведен список четырех относящихся к корпусу факторов, создающих определенные трудности при извлечении ключевых выражений. Этими факторами являются:

1. длина документа (длинные тексты содержат большее количество ключевых выражений-кандидатов). Авторы статьи делают вывод о том, что извлечение ключевых выражений из научных статей и разного рода текстов, связанных с документацией, приобретает более сложный характер, чем извлечение из аннотаций, электронных сообщений и новостей;
2. структура документа (текстовый документ может быть построен по определенной схеме – вступление, основная часть, заключение – а может иметь собственное строение, организованное автором). В данном случае авторы утверждают, что ключевые выражения извлекаются с большим успехом из научных статей и технической документации, поскольку эти типы текстов обладают стандартным форматом представления информации (аннотацией, вступлением, заключением и т.д.). Сложность могут представлять такие тексты, как веб-страницы, блоги, форумы и обзоры, поскольку их структурное строение не зависит от общепринятых стандартов, а только от авторского намерения выразить ту или иную мысль;
3. тема документа (ключевые выражения могут содержаться не только в начале [14], но и в конце документа [21]). В данном пункте авторы статьи подчеркивают обилие тем в текстах разговорного типа, диалогах, чатах и т.д.;
4. взаимосвязи тем (обычно ключевые выражения связаны друг с другом в одном тексте [22], [24]). Однако, это утверждение не всегда

верно для неформальных типов текстов (электронных сообщений, чатов, неформальных встреч, личных блогов и т.д.). Такие тексты могут содержать в себе множество несвязанных друг с другом тем, что будет затруднять процесс извлечения ключевых выражений.

Задача извлечения ключевых выражений весьма неоднозначна, поэтому самой большой сложностью является выбор метрики для достижения наиболее эффективного результата.

Далее приведем обзор существующих методов извлечения ключевых выражений, разработанных разными авторами с применением подходов, обозначенных выше.

Глава 2. Анализ и отбор методов автоматического извлечения ключевых выражений

В настоящем исследовании для проверки выдвинутой гипотезы были взяты разные алгоритмы автоматического извлечения ключевых выражений: статистические, гибридные, графовые, с машинным обучением и лингвистические. Важными параметрами работы алгоритмов являются возможность работы с русскоязычными текстами, выдача ключевых выражений в виде униграмм, биграмм, n-грамм и т.д. В данной главе алгоритмы объединены в подходы, перечисленные выше.

2.1. Методы, основанные на статистическом подходе

2.1.1. Логарифмическая функция правдоподобия

Мера ассоциации Log-likelihood (коэффициент правдоподобия) считается классическим показателем синтагматической связи между элементами коллокаций (в данном случае под коллокациями будут рассматриваться ключевые выражения). В традиционном понимании коллокацией считается «комбинация двух или более слов, имеющих тенденцию к совместной встречаемости» [4].

Первым шагом в оценке максимального правдоподобия является выбор распределения вероятностей, который, как предполагается, генерирует данные. Точнее, нам нужно сделать предположение о том, какой параметрический класс распределений используется, например, класс всех нормальных распределений или класс всех гамма-распределений. Каждый такой класс представляет собой семейство распределений, индексированных конечным числом параметров. Например, класс нормальных распределений – это семейство распределений, индексированных его средним значением $\mu \in (-\infty, \infty)$ и стандартным отклонением $\sigma \in (0, \infty)$. Данные выбирают конкретный элемент класса, закрепив

параметры. Полученные таким образом оценки параметров будут называться оценками максимального правдоподобия (1):

$$\log\big(f(x_i;\mu,\sigma^2)\big) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum(x_i-\mu)^2, \quad (1)$$

где x – число параметров, μ – среднее значение, σ – стандартное отклонение.

По эффективности метод приближается к tf-idf [18], однако не требует обработки целого корпуса документов.

2.1.2. TF-IDF

Мера tf-idf [18] (Term Frequency – Inverse Document Frequency) определяет, насколько хорошо данное выражение описывает документ внутри корпуса. Большой вес tf-idf получают слова с высокой частотой в конкретном документе и при этом с низкой частотой в других документах [17].

Tf (term frequency) – отношение числа вхождений слова к общему количеству слов документа. Оценивается важность слова в пределах документа (2).

$$tf(t,d) = \frac{n_i}{\Sigma_k n_k}, \qquad (2)$$

где n_i – число вхождений слова в документ, а Σ_k, n_k – общее число слов в документе.

Idf (inverse document frequency) – инверсия частоты, с которой некоторое слово встречается в документах коллекции. Уменьшает вес широкоупотребительных слов (3).

$$i\,df = \frac{\log|D|}{|d_i \supset t_i|}, \qquad (3)$$

где $|D|$ – количество документов в корпусе; $|(d_i \supset t_i)|$ – количество документов, в которых встречается t_i.

Таким образом, мера tf-idf является произведением двух сомножителей (4):

$$TF-IDF(t,d,D) = tf(t,d) * (idf(t,D)) \qquad (4)$$

2.1.3. Критерий Хи-квадрат

Метрика Хи-квадрат [29] не требует использования в дополнительных корпусах и годится для извлечения ключевых выражений из отдельных документов. Метод использует матрицу встречаемости: два слова встречаются вместе, если оба употреблены в одном предложении (отрезок текста, разделенный знаками пунктуации). Слова являются важными для документа, если они встречаются с другими словами в этом документе чаще, чем если бы они были распределены в нем случайно. Таким образом, для некоторого слова w_i это отношение встречаемости w_i и w_j к числу всех других сочетаний со словом w_i. Тогда высокое значение означало бы, что w_i – скорее всего, ключевое выражение для документа [20].

При извлечении ключевых выражений может возникнуть проблема с низкочастотными словами, которые получали бы неоправданно высокое значение, так как матрица была бы сильно разреженной. Чтобы исправить это, предлагается использовать критерий хи-квадрат (5):

$$x^2 = \sum [\frac{(O_i - F)_i^{\,2}}{E_i}], \quad (5)$$

где n – число слов, O – наблюдаемая частота, E – ожидаемая частота, F – теоретический закон распределения.

По эффективности метод приближается к tf-idf [18], однако не требует обработки целого корпуса документов.

2.1.4. YAKE

Статистический алгоритм YAKE (Yet Another Keyword Extractor) является одним из самых эффективных при извлечении ключевых выражений [12]. Он определяет набор из пяти функций, отражающих характеристики ключевых

выражений, которые эвристически объединены с целью присвоения единой оценки каждому ключевому выражению.

Особенностями данного экстрактора являются:

1. подход без учителя. Алгоритм извлечения основан на статистических характеристиках текста, алгоритму не требуется обучающий корпус.
2. независимость от корпуса. Алгоритм извлекает ключевые выражения из заданного ему текста без обращения к другим текстам.
3. независимость от языка. Алгоритм работает и с другими языками, помимо английского.
4. внутренний список стоп-слов.

Данный алгоритм обладает свойством автономности, поскольку работает при помощи машинного обучения без учителя. Алгоритм состоит из пяти этапов [13]: (1) предварительная обработка текста и определение терминов-кандидатов; (2) извлечение признаков; (3) подсчет веса термина; (4) создание n-грамм и вычисление веса кандидата в ключевое выражение; (5) дедупликация и ранжирование данных.

Выдача программы организована так, что, чем ниже оценка, тем более подходящее выражение было извлечено. Недостатком, однако, является выдача слов в парадигме, использованной в тексте.

2.2. Методы, основанные на лингвистическом подходе

2.2.1. Инструмент PullEnti

Система лингвистического процесса PullEnti [27] включает в себя морфологического и семантико-синтаксического анализа для выделения ключевых выражений. Работа в данной системе осуществляется через демонстрационную версию прямо на указанном выше сайте или путем

скачивания пакетов программ на требуемом языке программирования или исполнительного файла с готовым интерфейсом.

В системе PullEnti используются динамически подключаемые плагины, что позволяет без перекомпилирования подключать различный функционал. Именно таким образом подключается блок семантического анализа.

Процесс анализа в системе происходит следующим образом:

1. выделение токенов, представляющих собой типизированные фразы (например, текстовые, числовые и т.д.). Кроме того, выделяются метатокены – сложные токены, объединяющие несколько простых токенов (существительное с определителями, кавычки, скобки и т.д.);
2. опционально существует статистический словарь терминов, который пополняется, а затем включённые в него термины проверяются на наличие в других текстах;
3. создание «аналитического контейнера» [8], в который помещаются выделяемые сущности, токены в определенной последовательности, статистика и т.д.;
4. морфологический анализ, в ходе которого определяются части речи словоформ, род, падеж, число, язык (так как система поддерживает работу на трех языках). Далее определяется, есть ли словоформа в словаре, осуществляется поиск варианта словоформы в именительном падеже единственного числа и проводится анализ дополнительной информации.

Таким образом, система построена для выявления разного рода информации из текстов, что как раз удовлетворяет задаче настоящей работы.

2.2.2. RAKE

Гибридный алгоритм RAKE (Rapid automatic keyword extraction), запатентованный авторами в 2009 году [25], основан на предположении, что

ключевые элементы текста часто состоят из нескольких слов, но при этом не содержат знаки пунктуации, служебные слова и слова, не несущие ярко выраженного лексического значения. В исходном варианте данный алгоритм предназначен для работы с англоязычными текстами, однако, уже были предприняты действия по его модификации для работы с русскоязычным материалом [25].

Входными параметрами RAKE являются:

1. список стоп-слов;
2. набор разделителей (маркеров границ) для словосочетаний;
3. набор разделителей для слов.

При помощи набора разделителей текст делится на массив слов. Данный массив делится на последовательности смежных слов по заданным разделителям и позициям, в которых стоят стоп-слова. Слова в последовательности расставляются в том же порядке, как и в тексте, и все вместе рассматриваются в качестве кандидатов в ключевые выражения.

Для каждого кандидата в ключевые выражения строится таблица его встречаемости с другими словами, и вычисляется вес. Всего было разработано три метрики, основанные на степени и частоте:

1. частота слова – тем выше, чем чаще слово встречается, вне зависимости от количества смежных слов;
2. степень слова – тем выше, чем чаще слово встречается в тексте и чем длиннее словосочетания с ним;
3. отношение степени к частоте – тем выше, чем длиннее словосочетания.

Вес выражения вычисляется как сумма весов составляющих его слов.

В качестве ключевых выражений документа отбирается топ-список слов-кандидатов.

Алгоритм был реализован на многих языках программирования, в частности, для Python существует две вариации: использующая встроенные

функции языка и основанная на возможностях, предоставляемых библиотекой NLTK.

2.3. Методы, основанные на машинном обучении

2.3.1. TextRank

Алгоритмы ранжирования, основанные на графах, предполагают машинное обучение: присваивают значение вершине графа, используя общую информацию, рекурсивно собранную по всему графу [9].

Графовый алгоритм, применяемый к текстам на естественном языке, предполагает следующие процедуры:

1. вычленить единицы текста, соответствующие поставленной задаче, и добавить их к вершинам графа;
2. определить отношения между единицами и с их помощью провести ребра между вершинами графа;
3. применить алгоритм ранжирования;
4. отсортировать вершины по значениям.

В TextRank [28] в качестве отношений между вершинами используется совместная встречаемость с произвольной шириной окна. Вначале текст токенизируется, затем проводится частеречная разметка, применяется синтаксический фильтр и между словами, встречающимися в окне, добавляются рёбра. После добавления приблизительных значений PageRank каждой вершине они сортируются в обратном порядке и первые вершины подвергаются пост-обработке.

2.3.2. KeyBERT

Модель BERT [16], предложенная разработчиками компании Google, является двунаправленным трансформером, который позволяет

преобразовывать предложения и документы в векторы, которые отражают их значение.

В основе метод KeyBERT [19] лежит основная идея самой модели BERT. Создателем данного метода является Maarten Grootendorst. Данный подход определяется как метод, использующий машинное обучение с учителем.

Алгоритм реализован несколькими стадиями. Для начала создается список кандидатов ключевых выражений. В данном случае используется алгоритм CountVectorizer из библиотеки Scikit-Learn в среде программирования Python. CountVectorizer позволяет определить длину ключевых выражений и определять эти выражения как ключевые.

Затем сам документ и полученные ключевые выражения из него конвертируются в векторы при помощи модели BERT. Далее к полученным данным применяется пакет *sentence-transformers* для создания векторов слов. Существует большое количество предобученных моделей BERT, автор метода KeyBERT использует *distilbert — base-nli-stsb-mean-tokens* так как эта модель показала высокие результаты в оценке семантической близости.

Заключающим этапом алгоритма является поиск наиболее близких к тексту кандидатов. Предполагается, что хорошими ключевыми выражениями являются те выражения, которые больше всего похожи на сам текст, то есть эти слова могут представить сам текст. Чтобы вычислить сходство между текстом и выражениями, используется косинусное сходство между векторами. Выбор обусловлен тем, что это сходство хорошо работает в высокой размерности.

2.3.3. Topia

Статистический алгоритм Topia [30] определяет важные для текста термины, используя лингвистические инструменты (разметка частей речи и простой статистический анализ для определения терминов и их силы).

Алгоритм начинает работу с токенизации, затем проводится разметка в два этапа: терминам присваивается тег на основе лексикона и нормализованной

формы самого термина, после чего к каждому размеченному термину применяется набор правил. Правила устанавливаются согласно грамматическому строю данного языка.

Извлечение ключевых слов происходит по параметру количества вхождений слова в текст. Ключевые словосочетания же извлекаются несколько иначе, для них существует специальный параметр силы, определяющий количество слов в термине. То есть кандидатом на ключевое словосочетание по умолчанию являются термины с силой больше 1 независимо от количества вхождений.

Для русского языка реализована библиотека rutermextract [26], использующая морфологический анализатор pymorphy2 [23] для морфологического анализа. Библиотека работает на правилах, как и англоязычный экстрактор Topia. Извлеченные ключевые выражения приводятся в лемматизированной форме и упорядочиваются от более важных к менее важным.

Опциональными аргументами при вызове библиотеки являются: количество слов, семантическое «гнездо», то есть ключевые выражения, лежащие внутри других ключевых слов (например, вместе с «функциональный язык программирования» извлекаются «язык программирования» и «программирование»), вес (в данную функцию передается объект типа *Term*, по умолчанию ключевые выражения упорядочиваются по количеству употреблений, затем по количеству слов), строчная выдача.

Глава 3. Лингвистические данные для проведения экспериментов

Одной из первых задач на пути выполнения практического задания было составление пользовательского корпуса. Для достижения наиболее эффективных результатов, отвечающих требованиям поставленной гипотезе, корпус должен соответствовать следующим положениям:

1. содержать в себе тексты разных функциональных стилей (художественный, научный, публицистический);
2. содержать в себе тексты с аннотацией (у публицистического подкорпуса - развёрнутые заголовки статей, у научного - непосредственно аннотация, у художественного - выделенные автором сюжетные составляющие каждой главы).

Таким образом, тексты подбирались в соответствии с указанными выше требованиями. Тексты для корпуса собирались вручную с сайтов, где невозможно было применить автоматическую сборку, например, файлы-документы в формате .pdf. Тексты с новостных порталов собирались при помощи библиотеки Beautiful Soup в Python (приложение А).

Корпус состоит из трех подкорпусов и разделен на файлы в формате .txt, файлы разделены в соответствии с функциональным стилем. На данный момент корпус насчитывает 1 миллион слов до обработки.

Были использованы следующие источники для подкорпусов:

1) публицистический подкорпус:
   а) новостные порталы https://paperpaper.ru/ и https://meduza.io/;
2) научный подкорпус:
   а) сборники научных статей по компьютерной лингвистике разных лет;
3) художественный подкорпус:
   а) юмористическая повесть Джерома К.Джерома «Трое на четырех колесах» (перевод М. Жаринцовой);

b) юмористическая повесть Джером К.Джерома «Трое в одной лодке, не считая собаки» (перевод М.Салье);
c) роман-робинзонада Жюля Верна «Таинственный остров» (перевод Наталия Немчинова, Анна Худадова).

Собранный и отвечающий требованиям корпус был подвержен графематическому анализу, направленному на выделение нетекстовых элементов (таблиц, рисунков, формул, гиперссылок, числовых данных и пр.).

Глава 4. Методика проведения исследования

4.1. Проблемы извлечения ключевых выражений экспертным и компьютерным способами

Извлечение ключевых выражений возможно несколькими способами – вручную и автоматически. Экспертный подход имеет под собой, как преимущества, так и недостатки. Главным недостатком экспертного подхода является тот факт, что ключевые выражения выделяются людьми интуитивно [2]. Данное положение дел снижает степень научности такого подхода.

Еще одним «минусом» ручного извлечения ключевых выражений является «отсутствие корреляции между размером текста и количеством ключевых слов» [3]. Согласно исследованию [6], ключевые выражения должны составлять в тексте 25-30%. При извлечении ключевых выражений человеком это требование зачастую нарушается.

Немаловажным к тому же является обстоятельство того, что выделенные вручную ключевые выражения могут вовсе не присутствовать в тексте, поскольку эксперт склонен придумывать слова, подыскивать синонимы, отражающие его субъективное восприятие прочитанного. Эта проблема чаще всего возникает при работе с текстами художественной литературы.

Однако, преимуществами ручного извлечения ключевых выражений является извлечение концептов и обобщений написанного. Так, например, осуществляется присвоение тегов для новостных текстов, когда из текста берутся концепты ключевых выражений для облегчения поиска статей на интересующую тему.

Компьютерное извлечение ключевых выражений полностью исключает эмоциональную и субъективную составляющие, а также подстановка новых выражений становится полностью невозможной.

Машинное извлечение в данном случае кажется более научным, поскольку данные о частотности употребления выражения и степени его «keyness» свидетельствуют о доказательности и объективности показателей.

С другой стороны, автоматическое извлечение влечет за собой проблему чрезмерной «формализации». Так, например, в случае неиспользования стоп-словаря с включенными в него служебными словами, эти слова определяются более значимыми, чем знаменательные ввиду их частой повторяемости.

Таким образом, возникает проблема двух подходов: «эмоциональность» экспертного подхода, что влечет за собой некорректность получаемых данных, и чрезмерная «формализация» автоматического извлечения, выражающаяся в приоритете служебных частей речи, не способствующих раскрытию тематики текста и не составляющих его информационный портрет.

4.2. Методика проведения экспериментов

Характеристиками, по которым можно вычислить информативность ключевых выражений, являются частота слов и их расположение в документе. Под частотой слова подразумевается количество вхождений выражения в тексте. Чем больше вхождений выражения в документ, тем выше его информативность. Однако, в таком случае необходимо использование стоп-словаря с часто встречающимися не содержащими какой-либо информации о тексте словами.

Что касается расположения выражений в тексте, здесь важно учитывать место в тексте, где используется важное и уникальное для текста выражение. Часто информативные выражения употребляются в начале текста, аннотации и заголовке.

Настоящая работа нацелена на исследование зависимости степени важности ключевого выражения в тексте от места его расположения. Для проведения эксперимента были использованы русскоязычный корпус и набор описанных выше методов. Обоснованием выбора перечисленных в предыдущей главе методов являются их возможность работы с русскоязычными текстами и широкое употребление для задачи автоматического извлечения ключевых выражений.

Для данного исследования был собран русскоязычный корпус, состав которого описан в главе 3. Корпус состоит из трех подкорпусов разных функциональных стилей: публицистического с текстами новостей, научного со статьями о компьютерной лингвистике и художественного с текстами романов мировой классической литературы. Тексты в корпусе не размечались. Из каждого подкорпуса были взяты 50 текстов (50 новостей, 50 научных статей и 50 глав художественных произведений). Далее было взято начало каждого текста (для публицистического подкорпуса - заголовок и первые два-три предложения новости, для научного - аннотация и первый абзац статьи, для художественного - размеченные автором опорные слова и первый абзац главы). В сервисе Google-form был создан опрос, который был предложен информантам для ручной разметки ключевых выражений. Традиционная методика проведения эксперимента со стандартной инструкцией, предложенной А.С. Штерн [7]: «Прочитайте текст. Подумайте над его содержанием. Выпишите 10-15 слов, наиболее важных с точки зрения его смысла». Для проведения данного эксперимента в условие традиционной задачи были внесены некоторые правки – выделить 3-7 ключевых выражения, ранжировать их указав от самого важного к менее важному, а также были даны инструкции к тому, что есть ключевое выражение для каждого функционального стиля.

Определения ключевых выражений для подкорпусов текстов:

1. публицистический: ключевым выражением для новостных текстов является именная группа, наиболее ярко и полно отражающая суть обозначенной в заголовке текста;
2. научный: ключевым выражением для новостных текстов являются термины, именные группы использованных методов и т.п. авторами статьи, а также предмет и объект, обозначенные в статье;
3. художественный: ключевым выражением для новостных текстов являются обобщения описываемых автором событий.

В эксперименте приняли участие 5 информантов.

Далее была организована работа с оставшейся частью текста каждого файла. Задача была в том, чтобы извлечь ключевые выражения из середины и конца текста автоматически при помощи методов, описанных выше. Затем эксперимент заключался в сравнении полученных результатов ручного и автоматического извлечения ключевых выражений из разных частей текстов разных функциональных стилей.

Таким образом, параметры проводимых экспериментов приведены в таблице 1.

Таблица 1 – Параметры, применяемые для проведения экспериментов

<table>
<tr><th>Параметр</th><th colspan="2">Извлечение экспертами</th><th colspan="2">Извлечение алгоритмами</th></tr>
<tr><td rowspan="3">Объем текста</td><td>Публицистический</td><td>2-3 предложения</td><td>Публицистический</td><td>Максимум 10 предложений</td></tr>
<tr><td>Научный</td><td>Аннотация + первый абзац</td><td>Научный</td><td>Основная часть + заключение</td></tr>
<tr><td>Художественный</td><td>Опорные слова автора + первый абзац</td><td>Художественный</td><td>Оставшаяся часть главы</td></tr>
<tr><td>Длина ключевого выражения</td><td colspan="2">Ограничений нет</td><td colspan="2">Ограничения зависят от метода</td></tr>
<tr><td>Объем списка ключевых выражений</td><td colspan="2">3-7</td><td colspan="2">5-1000</td></tr>
<tr><td>Способ ранжирования ключевых выражений</td><td colspan="2">По убыванию важности</td><td colspan="2">По убыванию важности</td></tr>
</table>

4.3. Процедура сравнения методов автоматического выделения ключевых выражений

В ходе эксперимента были получены следующие данные: 150 текстов 3 разных функциональных стилей, каждый из которых поделен на 2 части. Ключевые выражения первой части текста размечены информантами, а также там присутствуют так называемые авторские ключевые выражения (об аннотированной природе текстов говорилось в главе 3). Из второй части текстов ключевые выражения выявлены автоматически. Такое деление текстов для ручного и автоматического извлечения ключевых выражений было сделано намеренно по причине того, что автоматическое извлечение ключевых выражений из первой части текста – это процедура с ожидаемо положительным результатом, в то время как распознавание их в оставшихся частях текста – это нетривиальная задача, решение которой как раз подтверждает роль тех ключевых выражений, которые выявляются в начале текста. Полученные данные могут быть пригодны для обработки при помощи подхода, основанного на машинном обучении.

Процедура сравнения результатов заключается в проверке совпадения ключевых выражений, выделенных автоматически из второй части текста, с выражениями, выделенными информантами из первой части.

В случае совпадения результатов работы алгоритма с работой информанта будет опровергаться гипотеза о наличии ключевых выражений только в начале текста. В случае совпадения результатов работы алгоритма с авторскими ключевыми выражениями будет проверяться эффективность самого метода.

В настоящем эксперименте не ожидается высоких показателей точности, полноты и f-меры ввиду сформулированной нами гипотезы и дизайна самого эксперимента. По сравнению с оценкой других лингвистических процедур точность, полнота и f-мера настоящего исследования ожидаются быть объективно низкими, приближающимися к 0. За положительный результат в

настоящем эксперименте считается как можно большее количество совпадений ключевых выражений, извлеченных алгоритмами и размеченных экспертами.

Глава 5. Проведение экспериментов и процедуры сравнения результатов экспертов и методов автоматического извлечения ключевых выражений

5.1. Публицистический подкорпус

Материалом для публицистического подкорпуса являются 50 текстов из новостных порталов https://paperpaper.ru/ и https://meduza.io/. Тексты характеризуются краткостью, точным изложением фактов и наличием вступительного предложения, призванным привлечь внимание читателя содержанием новости.

Пример текста публицистического подкорпуса:

«В Петербурге возобновили плановую вакцинацию детей. Ее приостанавливали из-за коронавируса": В Петербурге сняли запрет на плановую вакцинацию детей, введенный в начале апреля. Постановление главного санитарного врача опубликовано на сайте Роспотребнадзора.

Вакцинация взрослых пока остановлена. Как пояснили в комитете по здравоохранению, она проводится лишь по эпидемическим показаниям. Например, в поликлинике можно сделать прививку против клещевого энцефалита. Ранее Минздрав приостановил плановую вакцинацию детей и взрослых из-за коронавируса. Пояснялось, что решение не касается прививок новорожденным. Актуальные новости о распространении COVID-19 в городе читайте в рубрике «Бумаги» «Коронавирус в Петербурге».»

Экспертам было предложено выделить ключевые выражения из первого абзаца текста, а автоматическая процедура извлечения ключевых выражений проводилась с оставшейся частью текста.

В таблице 2 представлены ключевые выражения, выделенные экспертами, а в таблице 3 – алгоритмами.

Таблица 2 – Извлеченные экспертами ключевые выражения для первого текста публицистического подкорпуса

ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
вакцинация Роспотребнадзор коронавирус	Петербург, коронавирус, вакцинация, детская медицина	Вакцинация детей, сняли запрет, Петербург, Роспотребнадзор	возобновили вакцинацию детей	Сняли запрет на вакцинацию

Таблица 3 – Первые пять ключевых выражений, извлеченных алгоритмами для первого текста публицистического подкорпуса

Алгоритм	Ключевые слова
Log-likelihood	'актуальные 'новости 9.138999264580956, 'бумаги 'коронавирус 9.138999264580956, 'вакцинацию 'детей 9.138999264580956, 'городе 'читайте 9.138999264580956, 'здравоохранению 'проводится 9.138999264580956
Хи-квадрат	'актуальные 'новости 36.0, 'бумаги 'коронавирус 36.0, 'вакцинацию 'детей 36.0, 'городе 'читайте 36.0, 'здравоохранению 'проводится 36.0
TF-IDF	Коронавирус 0.35, вакцинацию 0.34, взрослых 0.22, энцефалита 0.18, COVID 0.18, клещевого 0.18, прививок 0.18, эпидемическим 0.18, прививку 0.15, поликлинике 0.14
PullEnti	Прививок против клещевого энцефалита, плановая вакцинация детей, министерство здравоохранения, вакцинация взрослых, вакцинация
YAKE	('вакцинация взрослых', 0.049109949149038906), ('взрослых пока остановлена', 0.1022363752613495), ('коронавирус в петербурге', 0.11648653464971856), ('вакцинация', 0.11827954849491486), ('остановлена', 0.11827954849491486)
RAKE	эпидемическим показаниям 4.0, клещевого энцефалита 4.0, плановую вакцинацию 4.0, актуальные новости 4.0, вакцинация взрослых 3.5
TextRank	коронавируса, коронавирус, вакцинация взрослых, вакцинацию, новорожденным
Topia	коронавирус 2, вакцинация взрослых 1, комитет 1, здравоохранение 1, эпидемический показания 1
KeyBERT	[('здравоохранению', 0.841), ('коронавируса', 0.814), ('эпидемическим', 0.8066), ('коронавирус', 0.7974), ('распространении', 0.7887)

Процедура сравнения проводилась следующим образом: результаты экспертной разметки сопоставлялись с результатами работы алгоритмов. Таким

образом, были выявлены следующие совпадения для исследуемых алгоритмов и экспертов, см. таблицы 4-12.

Таблица 4 – Количество совпадений извлеченных ключевых выражений в публицистическом подкорпусе, проанализированном экспертами и алгоритмом Rake

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	1	1	0	0	0
2	3	3	1	0	0
3	1	1	0	0	0
4	4	1	1	0	0
5	2	2	1	0	0
6	2	1	1	0	0
7	0	1	1	0	0
8	0	0	1	0	0
9	1	1	1	0	0
10	2	0	2	0	0
11	2	1	0	0	0
12	3	2	0	0	0
13	0	0	0	0	0
14	1	0	0	1	0
15	0	0	0	1	0
16	2	1	2	0	1
17	1	1	1	0	0
18	3	2	3	0	1
19	3	1	1	1	1
20	1	0	0	0	0
21	0	1	0	0	0
22	0	0	0	0	0
23	3	4	4	2	1
24	2	2	1	3	1
25	1	0	0	0	0
26	1	2	1	0	0
27	0	1	0	0	0
28	0	1	0	0	1
29	3	3	0	1	0
30	1	2	0	0	0
31	1	0	0	1	0
32	0	2	0	0	0
33	1	1	0	0	0
34	1	1	1	0	0
35	2	0	2	1	0

Продолжение таблицы 4

36	0	0	0	0	0
37	2	0	2	1	0
38	1	2	1	0	0
39	0	0	1	0	0
40	1	1	1	1	0
41	2	2	0	2	0
42	0	0	0	0	0
43	2	1	1	0	0
44	2	1	1	0	0
45	1	1	1	0	0
46	2	2	1	0	0
47	0	0	0	0	0
48	1	1	1	0	0
49	2	2	0	0	0
50	2	2	1	3	0

Ключевыми выражениями публицистического подкорпуса, извлеченные алгоритмом Rake, являются как отдельные выражения, так и n-граммы. В списках преобладают существительные, однако, встречаются глаголы и наречия. Процедура проверки совпадений осуществлялась при полном совпадении n-граммы алгоритма и варианта эксперта. Rake показал достаточно высокий результат, в среднем на один текст приходится 1 точное совпадение из ключевых выражений экспертов и алгоритма.

Таблица 5 – Количество совпадений извлеченных ключевых выражений в публицистическом подкорпусе, проанализированном экспертами и алгоритмом KeyBert

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	2	2	0	0	0
2	1	1	0	0	0
3	0	1	0	0	0
4	1	1	0	0	0
5	1	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	1	0	0
9	0	0	0	0	0
10	0	0	0	0	0

Продолжение таблицы 5

11	0	0	1	0	1
12	1	1	0	0	0
13	0	0	0	0	0
14	1	0	0	0	0
15	0	0	0	0	0
16	0	0	0	0	0
17	0	0	0	0	0
18	1	0	0	0	0
19	0	0	0	0	0
20	0	0	0	0	0
21	0	0	0	0	0
22	0	0	0	0	0
23	0	0	0	0	0
24	0	1	0	0	0
25	0	0	0	0	0
26	1	0	1	0	0
27	0	0	0	0	0
28	0	1	0	0	0
29	0	0	0	0	0
30	1	1	0	0	0
31	0	0	0	0	0
32	0	0	0	0	0
33	0	0	0	0	1
34	0	1	0	0	0
35	0	0	0	0	0
36	0	0	0	0	0
37	1	0	0	0	0
38	0	0	0	0	0
39	0	0	0	0	0
40	0	0	0	0	0
41	0	1	0	0	0
42	0	0	0	0	0
43	0	0	0	0	0
44	0	0	0	0	0
45	1	0	0	0	0
46	0	0	0	0	0
47	0	0	0	0	0
48	0	0	1	1	0
49	1	1	1	1	0
50	0	1	0	0	0

Трансформером KeyBert извлекаются ключевые, количество n-грамм регулируется пользователем. В данной работы были извлечены униграммы, биграммы и триграммы. Ключевые выражения для публицистического подкорпуса получились более информативными и точными, чем для научного и художественного. Алгоритм обеспечил выдачу существительных, наиболее точно описывающих суть второй половины новости. Однако в некоторых случаях трансформер извлекает глаголы и глагольные группы, которые отсутствуют в результатах экспертов. Таким образом, среднее количество совпадений KeyBert и экспертов составляет 0,26 слова на один текст.

Таблица 6 – Количество совпадений извлеченных ключевых выражений в публицистическом подкорпусе, проанализированном экспертами и алгоритмом Topia

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	0	0	0	0	0
2	2	3	1	0	0
3	0	1	0	0	0
4	1	0	0	0	0
5	1	2	0	0	0
6	2	2	1	0	0
7	0	2	1	0	0
8	1	0	1	0	0
9	0	0	1	0	0
10	1	0	1	0	0
11	2	1	0	0	0
12	1	1	0	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	1	1	0	1	0
16	1	1	1	0	0
17	0	0	0	0	0
18	2	2	2	0	0
19	1	0	1	1	0
20	0	0	0	0	0
21	0	1	0	0	0
22	1	1	0	0	0
23	2	2	2	0	0
24	1	0	1	0	0
25	1	0	0	0	0

Продолжение таблицы 6

26	2	2	1	0	0
27	1	1	0	0	0
28	0	1	0	0	0
29	2	1	0	0	0
30	1	1	1	0	0
31	1	0	0	1	0
32	0	2	0	0	0
33	0	0	1	0	0
34	0	1	0	0	0
35	2	0	1	0	0
36	0	0	1	0	0
37	2	0	2	2	0
38	1	2	1	0	0
39	0	0	0	0	0
40	0	0	0	0	0
41	1	1	0	1	0
42	0	0	0	0	0
43	1	0	1	0	0
44	1	0	0	0	0
45	1	1	1	0	0
46	1	4	1	0	0
47	0	1	1	0	0
48	2	2	1	0	0
49	2	2	0	0	0
50	2	1	1	1	0

Алгоритм Topia организует выдачу десяти n-грамм, состоящих преимущественно из именных групп, поэтому глаголы, извлеченные экспертами, сразу же не рассматривались, как кандидаты на совпадение. Однако, даже с таким условием алгоритм набирает достаточно большое количество совпадений, и в среднем на один текст публицистического подкорпуса получается 0,88 совпадений с разметкой экспертов.

Таблица 7 – Количество совпадений извлеченных ключевых выражений в публицистическом подкорпусе, проанализированном экспертами и алгоритмом Yake

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	1	1	0	0	0
2	1	1	0	0	0

Продолжение таблицы 7

3	0	1	0	0	0
4	0	0	0	0	0
5	1	2	1	0	0
6	1	1	0	0	0
7	0	1	1	0	0
8	0	0	0	0	0
9	1	1	0	0	0
10	1	0	1	0	0
11	1	1	0	0	0
12	1	1	0	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	1	0
16	1	1	1	0	0
17	0	0	0	0	0
18	2	1	2	0	0
19	0	0	0	0	0
20	0	0	1	0	0
21	2	1	0	0	0
22	0	0	0	0	0
23	1	1	1	0	0
24	1	0	1	1	0
25	1	1	1	1	0
26	2	2	2	0	0
27	0	0	0	0	0
28	0	0	0	0	0
29	0	0	0	0	0
30	1	1	1	0	0
31	1	0	0	1	0
32	0	1	0	0	0
33	0	0	0	0	0
34	0	0	0	0	0
35	0	0	0	0	0
36	0	0	0	0	0
37	0	0	1	0	0
38	1	1	1	0	0
39	0	0	0	0	0
40	0	0	1	1	0
41	2	2	0	2	0
42	1	0	1	0	0
43	1	0	1	0	0

Продолжение таблицы 7

44	2	1	1	0	0
45	2	1	1	0	0
46	0	2	1	0	0
47	1	0	1	0	0
48	1	0	0	0	0
49	1	1	0	1	0
50	3	3	1	2	1

Ключевые выражения алгоритма Yake представляют собой разнообразное сочетание n-грамм, состоящих как из знаменательных частей речи, так и из служебных. Алгоритм предоставляет достаточно широкий выбор ключевых выражений для небольших новостей публицистического подкорпуса. Совпадения с разметкой экспертов довольно часты, в среднем на одну новость и одного эксперта приходится 0,7 слова.

Таблица 8 – Количество совпадений извлеченных ключевых выражений в публицистическом подкорпусе, проанализированном экспертами и алгоритмом TextRank

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	2	2	0	0	0
2	3	3	1	0	0
3	1	1	0	0	0
4	1	0	0	0	0
5	2	1	0	0	0
6	0	0	0	0	0
7	0	1	1	0	0
8	1	0	0	0	0
9	1	1	2	0	0
10	1	0	0	0	0
11	2	1	0	0	0
12	2	2	0	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	0	0
16	1	1	1	0	0
17	0	0	1	0	0
18	3	2	3	0	0
19	1	0	1	1	0
20	0	0	0	0	0

Продолжение таблицы 8

21	1	1	1	0	0
22	1	1	0	0	0
23	2	2	3	0	0
24	1	0	0	0	0
25	2	1	1	1	0
26	1	1	1	0	0
27	1	2	0	0	0
28	0	0	0	0	0
29	4	2	0	0	0
30	3	2	1	0	0
31	0	0	0	0	0
32	0	1	0	0	0
33	1	1	1	0	0
34	1	1	0	0	0
35	2	0	0	0	0
36	0	0	1	0	0
37	2	0	1	1	0
38	1	1	1	0	0
39	0	0	0	0	0
40	0	0	0	0	0
41	3	2	0	3	0
42	1	0	1	0	0
43	1	0	1	0	0
44	2	1	1	0	0
45	1	1	1	0	0
46	0	3	1	0	0
47	0	0	1	0	0
48	2	2	1	0	0
49	2	1	0	0	0
50	2	1	1	1	0

Графовый алгоритм TextRank для публицистического подкорпуса показал самые лучшие результаты в совпадениях с экспертами. Ключевыми выражениями алгоритма являются существительные, словосочетания из двух существительных, существительного и прилагательного, иногда в выдаче присутствуют глаголы. Для сравнительно небольших текстов новостей публицистического подкорпуса алгоритм справился очень хорошо, результаты

более информативные, чем для других двух подкорпусов. Максимальное среднее совпадений для алгоритма составляет 1,16 слова на текст.

Таблица 9 – Количество совпадений извлеченных ключевых выражений в публицистическом подкорпусе, проанализированном экспертами и алгоритмом tf-idf

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	2	2	0	0	0
2	2	2	0	0	0
3	1	1	0	0	0
4	2	1	0	0	0
5	2	2	1	0	0
6	1	1	1	0	0
7	0	1	1	0	0
8	1	0	0	0	0
9	2	2	2	0	0
10	0	0	0	0	0
11	2	1	0	0	0
12	2	2	0	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	0	0
16	1	1	1	0	0
17	0	0	0	0	0
18	3	2	4	0	0
19	1	0	1	1	0
20	0	0	0	0	0
21	1	0	1	1	0
22	1	1	0	0	0
23	2	3	3	0	0
24	2	0	1	1	0
25	2	1	1	1	0
26	2	2	1	0	0
27	1	1	0	0	0
28	0	0	0	0	0
29	4	2	0	0	0
30	3	2	1	0	0
31	0	0	0	0	0
32	0	2	0	0	0
33	0	0	1	0	0
34	1	0	0	0	0
35	2	0	1	0	0

Продолжение таблицы 9

36	0	0	1	0	0
37	1	0	1	1	0
38	2	1	1	0	0
39	0	0	0	0	0
40	1	1	0	0	0
41	2	1	0	2	0
42	1	0	1	0	0
43	1	0	1	0	0
44	2	1	1	0	0
45	1	1	1	0	0
46	1	3	3	0	0
47	1	1	1	1	0
48	1	1	1	0	0
49	4	2	1	1	0
50	2	3	1	2	0

Алгоритм tf-idf извлекает все части речи с очень небольшим количеством глаголов, однако, это не помешало иметь большое количество совпадений с мнениями экспертов. Униграммы, характерные для экспертной аннотации, имеют частое совпадение с алгоритмом. Tf-idf для публицистического подкорпуса имеет довольно большое среднее совпадений с экспертами – 1,26 слова на один текст.

Таблица 10 – Количество совпадений извлеченных ключевых выражений в публицистическом подкорпусе, проанализированном экспертами и алгоритмом chi-square

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	0	0	1	0	0
2	0	1	1	0	0
3	0	1	0	0	0
4	0	0	0	0	0
5	0	1	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	1	1	0	0
9	0	0	0	0	0
10	2	2	0	0	0
11	0	0	0	0	0
12	0	0	0	0	0

Продолжение таблицы 10

13	0	0	0	0	0
14	0	0	1	0	0
15	0	0	0	0	0
16	1	1	1	0	0
17	1	1	1	0	0
18	0	1	0	0	0
19	0	0	0	0	0
20	0	0	1	0	0
21	0	0	0	0	0
22	0	0	0	0	0
23	2	2	2	2	1
24	2	0	2	1	1
25	1	1	1	1	0
26	0	1	0	0	0
27	0	1	0	0	0
28	0	0	0	0	0
29	2	2	0	0	0
30	2	2	2	1	1
31	1	0	0	1	0
32	0	0	0	0	0
33	1	1	1	2	0
34	0	1	1	0	0
35	0	0	0	0	0
36	0	0	0	0	0
37	0	0	1	1	0
38	0	0	0	0	0
39	0	0	0	0	0
40	1	1	2	2	0
41	0	0	2	0	0
42	0	0	0	0	0
43	0	0	0	0	0
44	0	0	0	0	0
45	0	0	0	0	0
46	0	0	0	0	0
47	0	1	0	0	0
48	0	0	0	0	0
49	1	1	0	1	0
50	0	1	0	0	0

Ключевые выражения метрики chi-square имеют под собой особенность – выдача биграмм. Данный факт автоматически исключает ключевые выражения-

униграммы из разметки экспертов как кандидатов на совпадение. Именно по этой причине среднее для chi-square получилось всего 0,34 слова на один текст. Однако сами биграммы chi-square вполне могут считаться качественными ключевыми выражениями для текстов публицистического подкорпуса. Сочетания биграмм этого алгоритма состоят из всех частей речи русского языка.

Таблица 11 – Количество совпадений извлеченных ключевых выражений в публицистическом подкорпусе, проанализированном экспертами и алгоритмом PullEnti

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	2	3	1	0	0
2	3	3	1	0	0
3	0	1	0	0	0
4	3	1	0	0	0
5	3	4	1	0	0
6	3	1	1	0	0
7	0	3	1	0	0
8	1	0	1	0	0
9	1	1	2	0	0
10	3	1	4	0	0
11	1	0	0	0	0
12	3	3	0	0	0
13	2	1	0	0	0
14	1	0	1	0	0
15	2	1	0	0	0
16	2	2	1	0	0
17	1	0	0	0	0
18	2	2	3	0	0
19	2	1	2	1	0
20	0	0	0	0	0
21	1	1	1	0	0
22	1	1	0	0	0
23	2	2	3	0	0
24	3	2	2	2	0
25	3	2	2	1	0
26	2	1	2	0	0
27	1	3	0	0	1
28	0	1	0	0	1
29	5	3	0	1	0
30	3	3	1	0	0

Продолжение таблицы 11

31	1	0	0	1	0
32	0	1	3	0	0
33	1	1	1	0	0
34	2	1	1	0	0
35	3	1	1	0	0
36	0	0	0	0	0
37	1	0	2	2	0
38	2	2	2	0	0
39	0	0	0	0	0
40	2	2	2	1	0
41	2	2	1	2	0
42	1	0	1	0	0
43	2	2	2	0	0
44	2	1	1	0	0
45	2	1	1	0	0
46	1	3	2	0	0
47	3	2	3	1	0
48	2	5	1	0	0
49	4	2	2	1	0
50	3	2	1	3	0

Сервис по извлечению ключевых выражений получил самое большое количество совпадений по всем подкорпусам. В частности, для публицистического подкорпуса среднее совпадений составляет 1,8, что превышает остальные результаты минимум в два раза. Ключевые выражения сервиса PullEnti или как они называются в самом сервисе – ключевые сущности – характеризуются наличием всех частей речи, извлечением именованных сущностей, а также словосочетаний, отобранных алгоритмом в качестве потенциальных кандидатов. Широкий выбор ключевых сущностей позволил получить большое количество совпадений с экспертами.

Таблица 12 – Количество совпадений извлеченных ключевых выражений в публицистическом подкорпусе, проанализированном экспертами и алгоритмом Log-likelihood

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	1	1	1	1	0
2	1	1	1	0	0

Продолжение таблицы 12

3	0	1	0	0	0
4	2	0	1	0	0
5	1	1	1	0	0
6	2	1	1	0	0
7	0	1	1	0	0
8	0	1	1	0	0
9	0	0	1	0	0
10	2	1	1	0	0
11	1	0	0	0	0
12	3	3	0	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	1	0
16	2	2	1	0	0
17	1	1	1	0	0
18	0	0	0	0	0
19	1	0	1	1	0
20	0	0	1	0	0
21	0	1	0	0	0
22	0	0	0	0	0
23	1	1	1	1	1
24	3	0	2	1	0
25	1	1	1	1	0
26	2	3	1	1	1
27	0	1	0	0	0
28	0	1	0	0	1
29	3	3	0	0	0
30	2	2	1	0	0
31	1	0	0	1	0
32	0	2	0	0	0
33	0	0	0	0	0
34	0	2	1	0	0
35	0	0	0	0	0
36	0	0	0	0	0
37	0	0	1	1	0
38	0	0	1	0	0
39	0	0	0	0	0
40	1	1	2	2	0
41	0	0	0	0	0
42	1	0	1	0	0
43	1	0	1	0	1

Продолжение таблицы 12

44	2	1	1	1	1
45	2	1	1	0	0
46	1	2	1	0	0
47	1	2	1	1	0
48	1	4	0	0	0
49	1	1	0	1	0
50	1	2	1	1	0

Метрика Log-likelihood организует выдачу биграмм разных частей речи, встречающихся в тексте. Таким образом, остальные n-граммы, кроме биграмм, не рассматриваются в качестве совпадений с мнениями экспертов. Однако, это не помешало получить достаточно большое количество совпадений. Среднее совпадений для публицистического подкорпуса и алгоритма Log-likelihood составляет 0,84.

Таблица 13 содержит результаты совпадений ключевых выражений, извлеченных автоматически и вручную экспертами. Приведены результаты сумм совпадений и среднее совпадений по каждому эксперту. Максимальные показатели получились у эксперта номер 1 с алгоритмом PullEnti, 90 совпадений и в среднем 1,8 совпадений слов на один текст. Минимальные показатели у эксперта номер 5 по нескольким алгоритмам: tf-idf, TextRank и Topia, показатели равны 0. Остальные алгоритмы показатели некоторое количество совпадений и находятся в пределах, заключенных между максимальным и минимальным показателями.

На рисунке 2 приведена диаграмма с полученными результатами, полученными в ходе процедуры совпадения ключевых выражений.

Таблица 13 – Результаты оценки совпадений работы алгоритмов и экспертного решения в публицистическом подкорпусе

Алгоритм		ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
Log-likelihood	сумма	42	45	31	15	5
	среднее	0,84	0,9	0,62	0,3	0,1
Хи-квадрат	сумма	17	24	21	12	3
	среднее	0,34	0,48	0,42	0,24	0,06

Продолжение таблицы 13

TF-IDF	сумма	63	47	35	11	0
	среднее	1,26	0,94	0,7	0,22	0
PullEnti	сумма	90	75	57	16	2
	среднее	1,8	1,5	1,14	0,32	0,04
YAKE	сумма	35	30	23	10	1
	среднее	0,7	0,6	0,46	0,2	0,02
RAKE	сумма	66	54	36	18	6
	среднее	1,32	1,08	0,72	0,36	0,12
TextRank	сумма	58	42	29	7	0
	среднее	1,16	0,84	0,58	0,14	0
KeyBERT	сумма	13	13	5	2	2
	среднее	0,26	0,26	0,1	0,04	0,04
Topia	сумма	44	43	27	7	0
	среднее	0,88	0,86	0,54	0,14	0

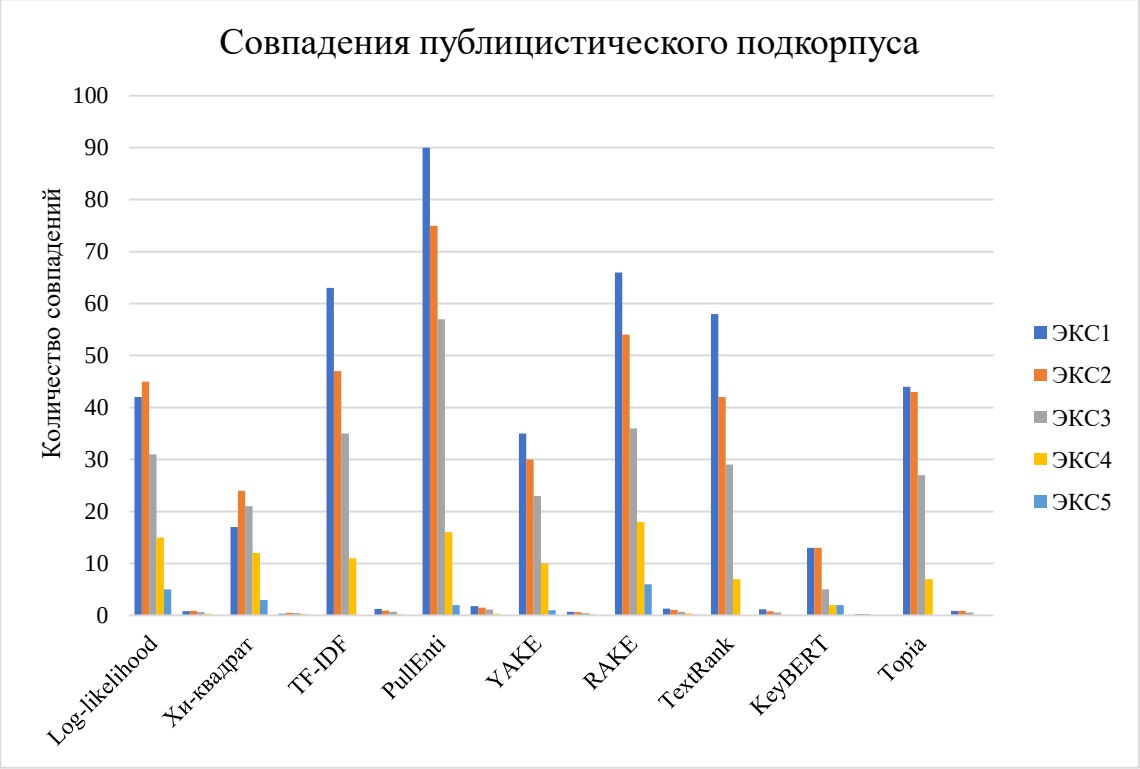

Рисунок 2. Диаграмма совпадений публицистического подкорпуса

5.2. Научный подкорпус

Материалом научного подкорпуса послужили 50 научных статей, посвященных компьютерной лингвистике из сборников научных статей разных лет. В каждой статье обязательно присутствуют название, аннотация, ключевые слова, определенные автором/авторами. Текст статьи соответствует классической структуре: введение, основная часть, заключение. Пример текста научного подкорпуса:

«И.В. Азарова, Е.Л. Алексеева

ОТ КРИТИЧЕСКОГО ИЗДАНИЯ К СТРУКТУРИРОВАННОМУ КОРПУСУ СЛАВЯНСКИХ ВАРИАНТОВ ЕВАНГЕЛИЯ

Аннотация. В статье рассматривается создание корпуса текстов на базе издания славянского Евангелия, включающего 28 рукописей, представляющих 8 групп славянских списков. Для обеспечения возможности поиска по разным типам текстовых фрагментов предлагается преобразование данных в корпус размеченных типизированных текстов.

Ключевые слова: Корпус, славянское Евангелие, Паратекст.

1. Проект издания славянского евангелия

Работа по проекту началась в 1993 г. при финансовой поддержке Немецкого библейского общества, выделившего средства для критического издания Евангелия от Иоанна, которое вышло в свет в 1998 г. Затем за счет Синодальной библиотеки Московского Патриархата было подготовлено издание Евангелия от Матфея, опубликованное в 2005 г. при поддержке РГНФ]. В настоящее время СПбГУ финансирует работу над Евангелиями от Марка и Луки и подготовку итогового издания в двух томах, содержащего критический текст всех четырех евангелий и результаты научного исследования материала.

План исследования славянского Евангелия был обоснован теоретически А.А. Алексеевым и описан в виде практической процедуры в.

Была проведена коллация двух фрагментов из Евангелия от Иоанна и Евангелия от Марка по 1500 рукописям с базовым текстом издания –

Мариинским Евангелием. Данные коллаций были сведены в структуру узлов разночтений, по которым был проведен автоматический кластерный анализ, позволивший выделить 8 неравных по объему групп. В издании представлены 28 рукописей как представители групп списков, что существенно упрощает нахождение чтений, которыми они противопоставлены друг другу, позволяет устанавливать генетические отношения между группами, реконструировать архетип, выдвигать текстологические гипотезы.

2. Структурированный корпус славянских текстов Евангелия

Проект издания Славянского евангелия находится на заключительной стадии: подготовлен материал для всего Четвероевангелия, однако изучение представленных в нем текстовых данных далеко от завершения. Неоднородность текста конкретного Евангелия, в котором редакционная правка зачастую охватывала лишь часть текста, требует дальнейшей систематизации рукописей и подтверждения стабильности выделенных групп. Чтобы обеспечить проверку итоговых данных издания, а заодно и положить начало новому этапу исследования рукописей, предполагается создание структурированного корпуса славянских вариантов Евангелия.

Каждый из списков, включенных в критический аппарат издания, представляется в корпусе в виде орфографически нормализованного текста, в котором будут обеспечены следующие варианты навигации по стихам соответствующих глав Евангелия: – изолированный просмотр текста каждой из 28 рукописей; – подстрочное представление базового Мариинского Евангелия и каждой из 28 рукописей; – представление узлов разночтений в виде структуры с гиперссылками с возможностью выбрать типы отображаемых разночтений из заданного списка: пропуск или вставка фрагмента текста, замена незнаменательного слова, замена знаменательного слова, перестановка фрагментов текста.

Преобразование текста издания в структурированный корпус предполагается выполнить с использованием программы Паратекст, которая предназначена для работы с библейскими текстами и содержит большое

количество маркеров, для спецификации фрагментов текста: как навигации по библейским текстам (по книгам, главам, стихам, фрагментам стихов), так и характеристике фрагмента для критического издания (пропуск, замена, вставка, перестановка элементов). Кроме того, программа Паратекст позволяет создавать подстрочные сопоставления текстов, проверять целостность текстов и облегчает издание, выполняя систематизированную разметку фрагментов текста для настольной издательской программы InDesign.»

Экспертам было предложено выделить ключевые выражения из первой части текста до абзаца, начинающегося со слов «План исследования…». Из второй части текста ключевые выражения извлекались автоматически. В таблице 14 представлены результаты разметки, а в таблице 15 – результаты алгоритмов для приведенной выше статьи.

Таблица 14 – Извлеченные экспертами ключевые выражения для первого текста научного подкорпуса

ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
Евангелие корпус поиск разметка	Корпус текстов, поиск текстовых фрагментов, Евангелие	Евангелия от Матфея, Проект издания славянского евангелия, СПбГУ, финансирует работу	Славянское Евангелие, создание корпуса, финансирование, поддержка	Текст четырех евангелий издается в двух томах

Таблица 15 – Первые пять ключевых выражений, извлеченных алгоритмами для первого текста научного подкорпуса

Алгоритм	Ключевые слова
Log-likelihood	'структурированный 'корпус 23.32921245239435, 'узлов 'разночтений 19.510127442625475, 'фрагментов'текста 16.641990085990116, 'каждой 'рукописей 16.599095782301788, 'славянского 'евангелия 13.79450086935657

Продолжение таблицы 15

Хи-квадрат	'автоматический 'кластерный 252.0, 'азарова е 252.0, 'алексеева 'план 252.0, 'алексеевым 'описан 252.0, 'анализ 'позволивший 252.0
TF-IDF	Евангелия 0.31, текста 0.22, рукописей 0.18, фрагментов 0.13, славянских 0.10
PullEnti	Текст, издание, рукопись, фрагмент, структурированный корпус
YAKE	('евангелия', 0.03194101703398826), ('обоснован теоретически', 0.043034937476037906), ('план исследования славянского', 0.0463618381902948), ('славянского евангелия', 0.05475769201358097), ('текста', 0.06127267736284642)
RAKE	подстрочное представление базового мариинского евангелия 20.833333333333332, проверку итоговых данных издания 13.666666666666666, стихам соответствующих глав евангелия 12.833333333333334, структурированный корпус славянских текстов 11.333333333333334, автоматический кластерный анализ 9.0
TextRank	Текста, евангелием, славянских текстов, издании, рукописей
Topia	структурированный корпус 3, замена 3, славянский евангелие 2, вид 2, евангелие 2
KeyBERT	[('противопоставлены', 0.8278), ('четвероевангелия', 0.8181), ('структурированного', 0.8116), ('автоматический', 0.8113), ('орфографически', 0.8109)]

Сравнение результатов работы алгоритмов и экспертов проводилось по процедуре полного совпадения ключевого выражения. Таким образом, были получены следующие совпадения для исследуемых алгоритмов и экспертов, см. таблицы 16-24.

Таблица 16 – Количество совпадений извлеченных ключевых выражений в научном подкорпусе, проанализированном экспертами и алгоритмом Rake

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	2	0	1	1	0
2	2	2	1	1	0
3	3	3	0	2	0
4	1	0	1	0	0
5	0	0	0	0	0
6	1	0	0	0	0
7	1	1	1	1	0
8	0	0	1	0	1

Продолжение таблицы 16

9	2	1	1	1	0
10	2	2	1	0	0
11	2	2	1	0	0
12	2	1	0	1	0
13	1	4	1	1	0
14	3	1	2	0	0
15	1	1	0	1	0
16	1	1	1	1	0
17	0	1	1	1	1
18	2	2	0	1	1
19	3	0	0	0	0
20	2	1	0	0	0
21	1	1	1	1	0
22	1	1	0	0	0
23	3	3	1	1	1
24	1	0	1	0	0
25	2	1	2	1	1
26	2	1	0	1	0
27	3	1	2	0	0
28	0	0	0	0	0
29	2	1	1	0	0
30	3	2	1	2	1
31	3	2	1	2	1
32	2	1	0	1	0
33	2	3	1	0	0
34	3	0	1	0	0
35	4	1	1	1	0
36	1	0	1	0	1
37	2	0	0	0	0
38	2	1	0	0	1
39	2	0	0	0	0
40	2	0	2	1	0
41	3	0	1	1	1
42	0	0	0	0	0
43	1	1	0	2	0
44	2	3	2	2	0
45	2	2	1	1	1
46	2	1	2	1	1
47	1	2	2	2	0
48	2	1	1	0	0
49	2	2	1	1	0

Продолжение таблицы 16

50	0	0	0	0	0

Ключевые выражения для научного подкорпуса, извлеченные алгоритмом Rake, характеризуются наличием только именных групп, состоящих из n-грамм. Ключевые выражения относятся к строгому научному стилю и состоят, прежде всего, из терминов или характерных для научного стиля словосочетаний, как, например, *новый этап исследования* и так далее. Ключевые выражения алгоритма Rake имеют частое совпадение с ключевыми выражениями экспертов, таким образом, среднее значение совпадений составляет 1,04 слова на один текст.

Таблица 17 – Количество совпадений извлеченных ключевых выражений в научном подкорпусе, проанализированном экспертами и алгоритмом KeyBERT

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	0	0	0	1	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0
11	0	1	0	0	0
12	0	0	0	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	0	1	0	1	0
16	0	0	1	0	0
17	0	0	1	0	0
18	0	0	0	0	0
19	0	0	0	0	0
20	1	1	0	0	0
21	0	0	0	0	0
22	0	0	0	0	0
23	0	0	0	0	0
24	0	0	0	0	0

Продолжение таблицы 17

25	1	1	1	0	0
26	0	0	0	0	0
27	0	0	0	0	0
28	0	0	0	0	0
29	1	1	0	0	0
30	0	0	0	0	0
31	0	0	0	0	0
32	0	0	0	0	0
33	0	0	0	0	0
34	0	0	0	0	0
35	0	0	0	0	0
36	0	0	0	0	0
37	0	0	0	0	0
38	0	0	0	0	0
39	0	0	0	0	0
40	0	0	0	0	0
41	0	0	0	0	0
42	0	0	0	0	0
43	0	0	0	0	0
44	0	0	0	0	0
45	0	0	0	0	0
46	0	0	0	0	0
47	0	0	0	0	0
48	0	0	0	0	0
49	0	0	0	0	0
50	0	0	0	0	0

Униграммы, биграммы и триграммы алгоритма KeyBERT преимущественно состоят из прилагательных, в редких случаях из сочетаний прилагательных и существительных, но поскольку трансформер извлекает выражения на основе предсказания следующего за ключевым выражением контекста, в результатах триграмм появляются глагольные группы. Таким образом, результаты KeyBERT не проявляют большого количества совпадений с ключевыми выражениями экспертов, и среднее слов на один текст составляет 0,06.

Таблица 18 – Количество совпадений извлеченных ключевых выражений в научном подкорпусе, проанализированном экспертами и алгоритмом Topia

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	1	1	0	1	0
2	2	2	0	1	0
3	0	0	0	1	0
4	0	0	1	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	2	0	1	1	0
8	0	0	0	0	0
9	2	1	2	1	0
10	1	2	1	0	0
11	0	0	0	0	0
12	0	0	1	0	0
13	0	0	0	0	0
14	3	1	2	0	0
15	0	1	0	1	0
16	1	0	1	0	0
17	0	0	1	0	0
18	1	0	0	0	0
19	1	0	0	1	0
20	2	1	0	0	0
21	0	0	0	0	0
22	1	0	1	0	0
23	2	1	1	0	0
24	1	0	1	0	0
25	1	1	1	1	1
26	1	1	0	1	0
27	0	0	0	0	0
28	0	0	0	0	1
29	2	1	0	0	0
30	2	1	0	1	0
31	4	1	3	0	0
32	1	1	0	1	0
33	1	2	1	0	0
34	3	0	1	0	0
35	2	1	1	0	0
36	0	0	0	0	0
37	1	0	0	1	0
38	1	0	1	0	0
39	0	0	0	1	0
40	1	0	0	0	0
41	1	0	0	0	0
42	0	0	0	0	0

Продолжение таблицы 18

43	0	0	0	0	0
44	3	0	1	2	0
45	0	0	0	0	0
46	1	1	2	0	0
47	1	2	2	1	0
48	0	0	0	0	0
49	1	0	1	0	0
50	0	0	1	0	0

Алгоритм Topia специализируется на извлечении терминов, поэтому результаты для этого подкорпуса получились четкие, понятные и часто встречающиеся у экспертов. Среди ключевых выражений Topia встречаются преимущественно униграммы, однако биграммы тоже есть среди выдачи. Относительно всех трех подкорпусов с научным Topia справился лучше всего, среднее количество совпадений алгоритма и результатов экспертов равно 0,94. Такой высокий коэффициент получился именно по тому, что у экспертов была тенденция выделять униграммы.

Таблица 19 – Количество совпадений извлеченных ключевых выражений в научном подкорпусе, проанализированном экспертами и алгоритмом Yake

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	1	1	1	1	0
2	2	2	0	1	0
3	0	1	0	1	0
4	0	0	0	0	0
5	0	0	1	0	0
6	2	0	0	0	0
7	2	1	2	1	0
8	1	0	2	0	0
9	3	2	2	1	0
10	2	1	1	0	0
11	0	0	0	0	0
12	0	0	1	0	0
13	0	0	0	0	0
14	3	1	2	0	0
15	0	0	0	0	0
16	0	0	0	0	0
17	0	0	1	0	0

Продолжение таблицы 19

18	0	0	0	0	0
19	1	0	0	1	0
20	2	2	0	0	0
21	0	0	0	0	0
22	1	0	1	0	0
23	2	1	1	1	1
24	1	0	0	0	0
25	1	1	2	0	0
26	2	1	0	0	0
27	0	0	0	0	0
28	1	0	0	0	1
29	2	1	1	0	0
30	1	0	0	0	0
31	3	1	2	0	0
32	1	1	0	1	0
33	1	1	0	0	0
34	3	0	1	0	0
35	1	0	1	0	0
36	0	0	0	0	0
37	1	0	0	1	0
38	1	0	0	1	0
39	0	1	0	1	0
40	1	0	0	0	0
41	1	0	0	0	0
42	0	0	0	0	0
43	1	1	0	1	0
44	2	0	0	2	0
45	1	0	0	0	0
46	1	1	2	0	0
47	1	2	2	1	0
48	1	0	1	0	0
49	2	0	1	0	0
50	0	0	0	0	0

В случае с алгоритмом Yake складывается похожая ситуация, однако выдача этого алгоритма отличается большим разнообразием в плане n-грамм. Среди ключевых выражений работы этого экстрактора есть и униграммы с самым большим весом, и биграммы, состоящие в основном из прилагательного с существительным, а также триграммы, в составе которых существительное,

управляющее падежной именной группой. Самое большое среднее совпадений Yake и результатами экспертов составляет 1,04.

Таблица 20 – Количество совпадений извлеченных ключевых выражений в научном подкорпусе, проанализированном экспертами и алгоритмом TextRank

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	1	1	0	0	0
2	2	2	0	1	0
3	0	1	0	1	0
4	1	0	0	0	0
5	1	0	0	0	0
6	2	0	0	0	0
7	2	0	0	0	0
8	0	0	0	0	0
9	1	0	1	0	0
10	1	1	1	0	0
11	1	0	0	0	0
12	1	1	2	1	0
13	0	0	0	0	0
14	1	0	1	0	0
15	1	0	0	0	0
16	1	0	0	0	0
17	0	0	0	0	0
18	0	0	0	0	0
19	1	0	0	1	0
20	1	0	1	0	0
21	0	0	1	0	0
22	1	0	1	0	0
23	3	1	1	1	0
24	0	0	0	0	0
25	0	0	0	0	0
26	1	0	0	0	0
27	0	0	0	0	0
28	0	0	0	0	0
29	1	0	0	0	0
30	0	0	0	0	0
31	4	0	3	0	0
32	2	1	0	1	0
33	0	0	0	0	0
34	3	0	1	0	0
35	1	0	1	0	0
36	0	0	0	0	0

Продолжение таблицы 20

37	1	0	0	1	0
38	0	0	0	0	0
39	0	0	0	1	0
40	3	0	1	0	0
41	1	0	0	0	0
42	0	0	0	0	0
43	1	1	0	1	0
44	0	0	0	1	0
45	1	0	0	0	0
46	0	0	0	0	0
47	1	0	0	0	0
48	1	0	0	0	0
49	1	0	1	0	0
50	0	0	0	0	0

Графовый алгоритм TextRank преимущественно извлекает униграммы-существительные. Недостатком алгоритма является извлечение одной и той же лексемы в разных парадигмах, поэтому количество слов автоматически сокращается. Однако на общий результат по совпадению с экспертами это не особо повлияло, и среднее совпадений составляет 0,88.

Таблица 21 – Количество совпадений извлеченных ключевых выражений в научном подкорпусе, проанализированном экспертами и алгоритмом tf-idf

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	2	1	0	0	0
2	2	2	0	1	0
3	1	2	0	2	0
4	1	0	1	0	0
5	0	0	0	0	0
6	2	0	0	0	0
7	2	0	1	0	0
8	0	0	1	0	0
9	1	0	2	0	0
10	2	2	0	0	0
11	1	0	0	0	0
12	0	0	1	0	0
13	0	0	0	0	0
14	3	1	2	0	0
15	0	0	0	0	0

Продолжение таблицы 21

16	1	0	0	0	0
17	0	0	0	0	0
18	1	0	1	0	0
19	0	0	0	1	0
20	1	0	0	0	0
21	0	0	0	0	0
22	1	0	1	0	0
23	3	1	1	1	1
24	0	0	0	0	0
25	1	1	1	1	1
26	3	1	0	1	0
27	0	0	0	0	0
28	0	0	0	0	0
29	1	1	1	0	0
30	0	0	0	1	0
31	3	0	3	0	0
32	2	1	0	1	0
33	0	0	0	0	0
34	3	0	1	0	0
35	2	0	1	0	0
36	0	0	0	0	0
37	1	0	0	1	0
38	1	1	0	0	1
39	0	0	0	0	0
40	2	0	1	0	0
41	0	0	1	0	0
42	0	0	0	0	0
43	0	0	0	1	0
44	3	0	1	2	0
45	2	0	0	0	0
46	0	0	0	0	0
47	0	0	0	0	0
48	0	0	0	0	0
49	0	0	0	0	0
50	0	0	0	0	0

Ключевые выражения алгоритма tf-idf представляют собой только униграммы существительных-терминов и прилагательных, использованных в научном тексте. Самое большое среднее совпадений результатов экстрактора и результатов экспертов составляет 0,96 слова.

Таблица 22 – Количество совпадений извлеченных ключевых выражений в научном подкорпусе, проанализированном экспертами и алгоритмом chi-square

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	0	0	0	0	0
2	1	0	1	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0
11	0	0	0	0	0
12	0	0	0	0	0
13	1	0	0	0	0
14	0	0	0	0	0
15	0	0	1	0	0
16	0	1	0	0	0
17	1	0	1	1	1
18	1	1	0	0	0
19	0	0	0	0	0
20	0	0	0	0	0
21	0	0	0	0	0
22	0	0	0	0	0
23	1	0	1	0	0
24	0	0	0	0	0
25	1	1	1	0	0
26	0	0	0	0	0
27	0	1	0	0	0
28	0	0	0	0	0
29	0	0	0	0	0
30	0	0	0	0	0
31	1	1	0	0	0
32	1	1	0	1	0
33	1	1	0	0	0
34	0	0	0	0	0
35	0	0	0	0	0
36	0	0	0	0	0
37	0	0	0	0	0
38	0	0	0	0	0
39	0	2	0	1	0

Продолжение таблицы 22

40	0	0	0	0	0
41	0	0	0	0	0
42	0	0	0	0	0
43	0	0	0	0	0
44	0	0	0	0	0
45	0	0	0	0	0
46	2	2	1	1	0
47	0	0	0	0	0
48	0	0	0	0	0
49	0	0	0	0	0
50	0	0	0	0	0

Метрика для извлечения коллокаций chi-square, результатом которой являются биграммы, состоит из сочетаний разных частей речи, которые, по мнению алгоритмов, считаются коллокациями. Среди этих результатов нашлось небольшое количество совпадений с результатами экспертов, поскольку эксперты выделяли преимущественно униграммы. Таким образом, среднее совпадений ключевых выражений chi-square с результатами экспертов равно 0,22, что является очень низким результатом относительно метрики log-likelihood.

Таблица 23 – Количество совпадений извлеченных ключевых выражений в научном подкорпусе, проанализированном экспертами и алгоритмом PullEnti

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	2	1	0	1	0
2	3	2	0	1	0
3	3	3	1	2	0
4	2	0	1	0	0
5	3	0	0	0	1
6	1	0	0	0	0
7	3	3	1	0	0
8	0	0	1	0	2
9	2	1	2	1	0
10	2	2	1	0	0
11	2	2	0	0	0
12	2	2	3	1	0
13	2	3	1	1	1
14	3	1	2	0	0

Продолжение таблицы 23

15	2	0	1	0	0
16	2	2	1	1	0
17	2	1	1	0	0
18	3	2	1	0	0
19	3	1	1	2	0
20	1	1	3	0	0
21	1	1	0	1	0
22	1	1	2	2	0
23	4	1	1	0	1
24	1	0	1	0	0
25	3	1	2	2	1
26	2	1	0	0	0
27	4	0	2	0	0
28	2	0	1	0	0
29	1	1	2	1	1
30	3	1	1	1	0
31	5	2	4	0	1
32	5	2	0	2	1
33	2	2	0	0	0
34	3	0	1	0	0
35	3	1	1	2	0
36	2	0	2	1	0
37	3	0	1	3	0
38	3	1	1	0	0
39	1	2	0	1	0
40	3	0	1	0	0
41	1	0	2	0	0
42	0	0	0	0	0
43	3	2	0	1	0
44	2	2	2	3	0
45	2	1	0	1	0
46	2	3	2	2	0
47	2	3	2	2	0
48	2	2	1	0	0
49	3	2	1	1	0
50	4	3	2	2	0

Самое большое среднее совпадений для всех подкорпусов получилось у сервиса PullEnti, и оно составляет 2,32 слова на один научный текст. Выдача ключевых сущностей данного сервиса состоит из самых разнообразных n-грамм,

что, безусловно, находит положительный отклик в результатах экспертов. Среди результатов экстрактора присутствуют практически все существительные, использующиеся в качестве терминов в научном тексте. А также в результатах находятся глаголы и именованные сущности, являющиеся кандидатами в ключевые выражения.

Таблица 24 – Количество совпадений извлеченных ключевых выражений в научном подкорпусе, проанализированном экспертами и алгоритмом Log-likelihood

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	2	2	1	2	0
2	2	1	0	2	0
3	0	1	0	0	0
4	0	0	0	0	0
5	2	0	1	0	1
6	0	1	0	1	0
7	2	1	2	1	0
8	1	0	1	1	0
9	2	2	2	1	1
10	2	3	1	1	0
11	1	1	1	0	0
12	0	0	0	0	0
13	3	2	1	1	0
14	3	1	1	1	0
15	2	1	1	1	0
16	1	1	1	1	0
17	1	0	2	1	1
18	1	1	1	1	1
19	0	0	0	0	0
20	2	2	0	1	0
21	0	0	0	0	0
22	1	0	0	1	0
23	1	1	1	1	1
24	1	0	1	0	0
25	3	1	2	1	1
26	2	1	0	1	0
27	1	0	0	0	0
28	1	0	0	0	0
29	1	1	0	0	0

Продолжение таблицы 24

30	2	1	0	1	0
31	3	1	1	0	0
32	2	1	1	1	0
33	2	2	2	1	0
34	3	0	1	1	0
35	1	0	0	0	0
36	0	0	0	0	0
37	0	0	0	1	0
38	1	1	1	0	0
39	1	3	0	1	0
40	1	0	1	0	0
41	2	0	1	0	0
42	0	0	0	0	0
43	1	1	0	2	0
44	2	1	2	3	0
45	1	1	0	1	1
46	2	2	1	1	0
47	1	3	2	1	0
48	2	1	1	1	0
49	3	1	1	1	0
50	2	1	1	1	0

Результатами выдачи метрики Log-likelihood являются биграммы, что могло повлиять на результат по количеству совпадений ключевых выражений алгоритма и экспертов, однако, самое большое среднее совпадений получилось равно 1,4, что может рассматриваться, как высокий результат для метрики с выдачей биграмм.

В таблице 25 приведена сводная таблица результатов совпадений по научному подкорпусу. Таблица содержит значения суммы и среднего совпадений работы алгоритмов и экспертов. Максимальные значения получились у эксперта номер 1 и экстрактора PullEnti, сумма совпадений ключевых выражений равна 116, среднее – 2,32 слова на один текст. Минимальные значения у эксперта номер 5 и алгоритмов TextRank и KeyBERT, результат равен 0. Результаты остальных алгоритмов и экспертов находятся в пределах максимального и минимального значений.

Рисунок 3 демонстрирует диаграмму с результатами, полученными после проведения процедуры совпадения ключевых выражений.

Таблица 25 – Результаты оценки совпадений работы алгоритмов и экспертного решения в научном подкорпусе

Алгоритм		ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
Log-likelihood	сумма	70	44	36	37	7
	среднее	1,4	0,88	0,72	0,74	0,14
Хи-квадрат	сумма	11	11	6	4	1
	среднее	0,22	0,22	0,12	0,08	0,02
TF-IDF	сумма	48	14	21	13	3
	среднее	0,96	0,28	0,42	0,26	0,06
PullEnti	сумма	116	62	56	38	9
	среднее	2,32	1,24	1,12	0,76	0,18
YAKE	сумма	52	23	28	15	2
	среднее	1,04	0,46	0,56	0,3	0,04
RAKE	сумма	87	54	39	33	12
	среднее	1,74	1,08	0,78	0,66	0,24
TextRank	сумма	44	9	16	10	0
	среднее	0,88	0,18	0,32	0,2	0
KeyBERT	сумма	3	5	3	2	0
	среднее	0,06	0,1	0,06	0,04	0
Topia	сумма	47	22	28	16	2
	среднее	0,94	0,44	0,56	0,32	0,04

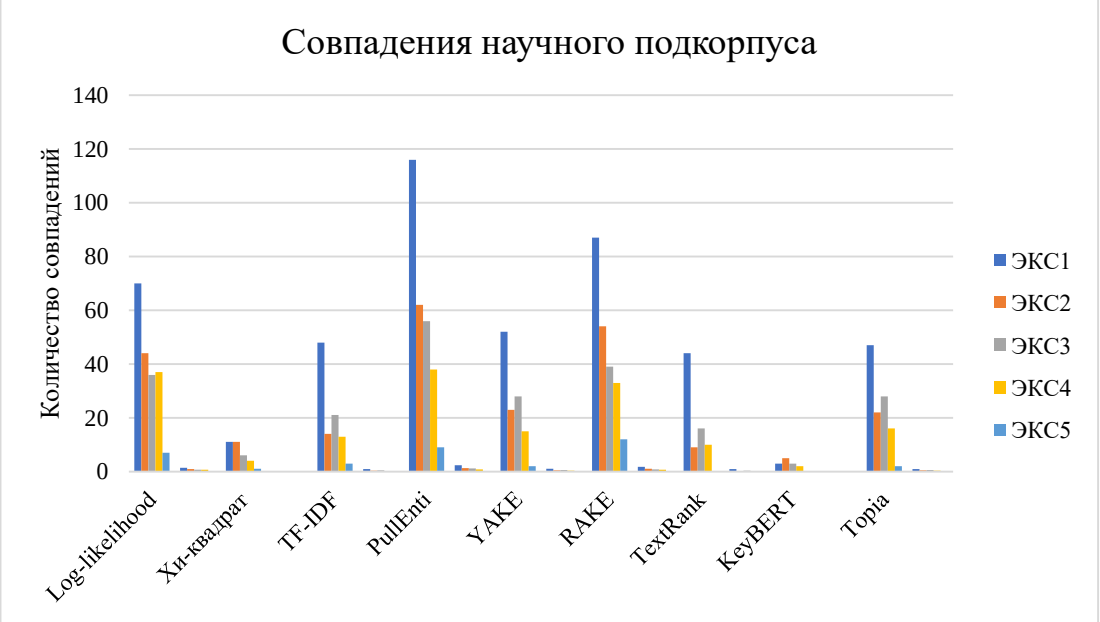

Рисунок 3. Диаграмма совпадений научного подкорпуса

5.3. Художественный подкорпус

В художественном подкорпусе использовались тексты двух писателей: Джерома К. Джерома и Жюля Верна. Каждая из 50 глав из их произведений начинается опорными словами, которые автор поместил перед самим текстом. Таким образом, читатель уже знакомится с некоторыми деталями сюжета и готовится к их раскрытию по ходу чтения главы. Пример текста художественного подкорпуса из произведения Джерома К. Джерома «Трое в лодке, не считая собаки» (в качестве примера взята шестнадцатая глава, так как она самая короткая по размеру):

«ГЛАВА ШЕСТНАДЦАТАЯ

Рэдинг. — Нас ведет на буксире паровой баркас. — Нахальное поведение маленьких лодок. — Как они мешают паровым баркасам. — Джордж и Гаррис снова уклоняются от работы. — Одна банальная история. — Стритли и Горинг.

Мы подъехали к Рэдингу часов около одиннадцати. Река в этом месте грязная и унылая. В окрестностях Рэдинга не хочется задерживаться надолго.

Рэдинг — старинный, знаменитый городок, основанный в далекие дни короля Этельреда, когда датчане поставили свои военные корабли в бухте Кеннет и, основавшись в Рэдинге, совершали набеги на Эссекс. Тут Этельред со своим братом Альфредом дали датчанам бой и разбили их, причем Этельред главным образом молился, а Альфред сражался.

В более поздние годы на Рэдинг, по-видимому, смотрели как на приятное местечко, куда можно было бежать, когда в Лондоне становилось скверно. Парламент обычно переезжал в Рэдинг всякий раз, как в Вестминстере объявлялась чума. В 1625 году юстиция последовала его примеру, и все заседания суда происходили в Рэдинге. На мой взгляд, лондонцам стоило претерпеть какую-нибудь пустяковую чуму, чтобы разом избавиться и от юристов, и от парламента.

Во время борьбы парламента с королем Рэдинг был осажден графом Эссексом, а четверть века спустя принц Оранский разбил под Рэдингом войска короля Иакова.

Генрих Первый похоронен в Рэдинге, в Бенедиктинском аббатстве, которое он сам основал и развалины которого сохранились до наших дней. В этом же самом аббатстве славный Джон Гонт был обвенчан с леди Бланш.

У Рэдингского шлюза мы поравнялись с паровым баркасом, принадлежащим одним моим знакомым, и нас подвезли на буксире почти до самого Стритли. Это очень приятно — идти на буксире за баркасом. Лично мне это нравится гораздо больше, чем гребля. Поездка была бы еще приятнее, если бы не множество маленьких лодчонок, которые все время сновали вокруг нашего баркаса. Чтобы не утопить их, нам то и дело приходилось замедлять ход и останавливаться. У этих весельных лодок пренеприятная привычка путаться на реке перед паровыми баркасами. Против них необходимо принять какие-то меры. И они к тому же еще такие нахальные, эти лодки. Чтобы они соблаговолили поторопиться, приходится так свистеть, что котел чуть не лопается. Будь на то моя воля, я бы время от времени топил парочку лодок, чтобы хорошенько их проучить.

Выше Рэдинга река становится очень приятной. У Тайлхерста ее несколько портит железная дорога, но от Мэплдерхэма до Стритли вид прямо великолепный. Несколько выше шлюза стоит Хардвик-Хаус, где Карл Первый играл в шары. Окрестности Пэнгборна, где находится прелестная гостиница «Лебедь», вероятно, столь же хорошо знакомы завсегдатаям картинных выставок, как и обитателям этой местности.

Мы отцепились от баркаса моих знакомых, немного не доезжая грота, и Гаррис принялся доказывать, что теперь моя очередь грести. Это показалось мне совершенно необоснованным. Утром мы условились, что я проведу лодку на три мили выше Рэдинга. Но ведь теперь мы были выше Рэдинга на десять миль! Конечно, грести надо было опять Гаррису и Джорджу.

Однако я не мог склонить ни того, ни другого к своей точке зрения и, чтобы не спорить напрасно, взялся за весла. Не успел я проработать и минуту, как мы увидели на реке какой-то черный предмет и приблизились к нему. Джордж наклонился и схватил этот предмет, но тотчас же с криком отшатнулся, бледный как полотно.

Это было тело мертвой женщины. Оно легко плыло по воде, и лицо утопленницы было кротко и спокойно. Его нельзя было назвать красивым, это лицо. Оно преждевременно состарилось, высохло и исхудало. Но это было милое и приятное лицо, несмотря на следы нужды и бедности, и на нем лежал отпечаток безмятежного спокойствия, которое мы часто видим на лице больных, когда их страдания наконец прекращаются.

На наше счастье, — нам вовсе не хотелось таскаться по судам и следователям — какие-то люди на берегу тоже заметили утопленницу и взяли на себя заботу о ней.

Впоследствии мы узнали историю этой женщины. Разумеется, это была обыкновенная, пошлая трагедия. Она любила и была обманута или сама обманулась. Так или иначе, она согрешила — это со многими из нас случается, — и ее знакомые и родные, охваченные справедливым негодованием, захлопнули перед ней двери своих домов.

Вынужденная бороться с судьбой одна, неся на шее ярмо своего позора, она опускалась все ниже и ниже. Сначала ей удавалось содержать себя и ребенка на двенадцать шиллингов в неделю, которые она получала, работая по двенадцать часов в день. Шесть шиллингов она платила за содержание ребенка, а на остальные пыталась кое-как удержать душу в теле.

Шесть шиллингов в неделю связывают тело с душой не слишком крепко. Соединенные столь хрупкими узами, они все время пытаются расстаться. И однажды, вероятно, несчастная особенно ясно увидела свою жизнь во всем ее тоскливом однообразии, со всеми ее страданиями, и насмешливая тень смерти испугала ее. В последний раз обратилась она за помощью к друзьям, но, оградившись ледяной стеной респектабельности, они не услышали голоса

отверженной. Тогда она съездила повидать своего ребенка, с каким-то тупым равнодушием взяла его на руки и поцеловала, не проявляя никаких чувств, и потом ушла, сунув ему в руку коробку грошовых конфет. На свои последние шиллинги она взяла билет и приехала в Горинг.

Как видно, горчайшие переживания ее жизни были связаны с лесистыми берегами и веселыми зелеными лужайками, окружающими Горинг. Но женщины почему-то любят гладить нож, который нанес им рану. А может быть, к горечи примешивались солнечные воспоминания о лучших часах, проведенных близ овеянных тенью струй, над которыми развесистые деревья так низко склоняют свои ветви.

Весь день пробродила она по лесу, что тянется вдоль берега реки. Потом, когда серые сумерки раскинули над водой свой темный плащ, она протянула руки к безмолвной реке, которая знала ее горести и радости. И старая река любовно приняла ее в объятия, прижала ее усталую голову к своей груди и успокоила боль. Так согрешила эта женщина во всем — и в жизни и в смерти. Мир праху ее и всех других грешников…

Горинг на левом берегу реки и Стритли на правом — очаровательные местечки, в которых приятно провести несколько дней. Воды реки до самого Пэнгборна так и манят поплавать в солнечный день под парусом или выехать в лунную ночь на лодке, а окружающий вид очень красив. Мы намеревались дойти в этот день до Уоллингфорда, но улыбка реки соблазнила нас остаться. Привязав лодку у моста, мы отправились в Стритли и позавтракали в гостинице «Бык» к великому удовольствию Монморенси.

Говорят, что горы, высящиеся на обоих берегах реки, когда-то соединялись и преграждали течение нынешней Темзы. Река будто бы оканчивалась несколько выше Горинга, образуя большое озеро. Я не могу ни подтвердить, ни опровергнуть это утверждение. Я просто отмечаю его.

Стритли — старинное местечко, основанное, как большинство прибрежных городов и поселков, во времена бриттов и саксов. В Стритли куда приятнее останавливаться, чем в Горинге, если у вас есть возможность

выбирать, но сам по себе Горинг достаточно красив и к тому же расположен ближе к железной дороге, что имеет значение, если вы хотите удрать из гостиницы, не заплатив по счету.»

В задачу экспертов входило выделить ключевые слова из начала текста до абзаца, начинающегося со слов «В более поздник годы…». Ключевые слова из оставшейся части текстов извлекались автоматическим способом. В таблице 26 приведены результаты работы экспертов, а в таблице 27 – результаты алгоритмов для приведенного выше текста.

Таблица 26 – Извлеченные экспертами ключевые выражения для шестнадцатого текста художественного подкорпуса

ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
Рэдинг Этельред набег	Рэдинг, Эльтеред, Альфред, история городка	Остановка по дороге, город, история	старинный, знаменитый городок	Рэдинг - старинный городок, основанный в дни короля Этельреда

Таблица 27 – Первые пять ключевых выражений, извлеченных алгоритмами для шестнадцатого текста художественного подкорпуса

Алгоритм	Ключевые слова
Log-likelihood	'шесть 'шиллингов 23.08830421221319, 'шиллингов 'неделю 23.08830421221319, 'баркасами 'них 14.841617637326035, 'бедности 'нем 14.841617637326035, 'бежать 'лондоне 14.841617637326035
Хи-квадрат	'баркасами 'них 615.0, 'бедности 'нем 615.0, 'бежать 'лондоне 615.0, 'безмятежного 'спокойствия 615.0, 'берегами 'веселыми 615.0
TF-IDF	Рэдинга 0.10, Река 0.08, Стритли 0.07, Горинг 0.07, баркаса 0.04
PullEnti	Текст, издание, рукопись, фрагмент, структурированный корпус

Продолжение таблицы 27

YAKE	('лондоне становилось скверно.парламент', 0.0011147190945592154), ('вестминстереобъявлялась чума', 0.007592783584426982), ('лондоне становилось', 0.007935821808963017), ('рэдинг всякий', 0.01837408138928375), ('поздние годы', 0.019128046883528284)
RAKE	руку коробку грошовых конфет 16.0, какую-нибудь пустяковую чуму 9.0, завсегдатаям картинных выставок 9.0, отпечаток безмятежного спокойствия 9.0, охваченные справедливым негодованием 9.0
TextRank	реке, рэдингом, стритли, баркасом, лодки
Topia	рэдинга 5, горинг 5, время 4, река 4, лодка 3
KeyBERT	[('вестминстере', 0.8833), ('городов', 0.8827), ('равнодушием', 0.8735), ('местечки', 0.8708), ('принадлежащим', 0.8707)]

Оценка совпадений проводилась по принципу сравнения ключевых выражений, извлеченных алгоритмами и размеченных экспертами. Итак, результаты совпадений приведены в таблицах 28-36.

Таблица 28 – Количество совпадений извлеченных ключевых выражений в художественном подкорпусе, проанализированном экспертами и алгоритмом Rake

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	4	0	2	0	0
2	2	0	0	0	0
3	3	0	0	0	0
4	2	1	0	0	0
5	3	0	1	2	0
6	3	1	0	1	0
7	3	0	1	0	0
8	1	0	1	0	0
9	2	1	0	0	0
10	1	1	2	0	0
11	3	0	1	1	0
12	2	2	1	0	0
13	2	1	1	0	1
14	3	2	0	0	0
15	1	1	0	0	0
16	1	1	1	0	1
17	2	2	2	0	0

Продолжение таблицы 28

18	2	1	0	1	0
19	3	1	1	0	1
20	2	2	2	1	1
21	1	0	0	0	0
22	2	1	1	0	0
23	0	1	1	1	0
24	1	0	1	0	0
25	1	2	2	1	0
26	1	0	1	1	1
27	3	2	1	2	0
28	1	0	0	0	0
29	0	0	1	0	0
30	2	0	1	0	0
31	2	1	2	0	0
32	1	0	0	0	0
33	0	0	0	0	0
34	1	1	0	0	0
35	1	0	0	0	0
36	1	1	1	0	0
37	1	1	0	0	0
38	0	0	1	0	0
39	0	1	1	0	0
40	1	2	1	2	0
41	2	2	2	0	0
42	3	2	0	2	0
43	1	2	1	0	0
44	1	1	1	0	0
45	0	0	0	0	0
46	0	1	0	0	0
47	0	1	0	0	0
48	2	2	1	1	1
49	1	0	0	0	0
50	1	0	0	1	0

Ключевые выражения алгоритма Rake предоставляют собой широкий спектр n-грамм, четко отражающих суть происходящего в повествовании опорных слов. Результаты данного алгоритма проявляют достаточное сходство с результатами экспертов, среднее совпадений на текст равно 1,52 слова. Среди ключевых выражений есть и названия предметов, их характеристика, описание

пейзажей, действий и участников событий. Таким образом, Rake раскрывает большую часть повествования при помощи извлеченных ключевых выражений.

Таблица 29 – Количество совпадений извлеченных ключевых выражений в художественном подкорпусе, проанализированном экспертами и алгоритмом KeyBERT

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	1	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0
11	1	0	1	0	0
12	0	0	0	0	0
13	0	0	0	0	0
14	1	0	0	0	0
15	0	0	0	0	0
16	0	1	1	1	0
17	0	0	0	0	0
18	0	0	0	0	0
19	0	0	0	0	0
20	1	1	0	0	0
21	0	1	0	0	0
22	0	0	0	0	0
23	0	0	0	0	0
24	0	0	0	0	0
25	0	0	0	0	0
26	0	0	0	0	0
27	0	0	0	0	0
28	0	0	0	0	0
29	0	0	0	0	0
30	0	0	0	0	0
31	0	0	0	0	0
32	0	0	0	0	0
33	0	0	0	0	0
34	0	0	0	0	0
35	0	0	0	1	0

Продолжение таблицы 29

36	0	0	0	0	0
37	0	0	0	1	0
38	0	0	0	0	0
39	0	1	0	1	0
40	0	0	0	0	0
41	0	0	0	0	0
42	0	0	0	0	0
43	0	0	0	0	0
44	0	0	0	0	0
45	0	0	0	0	0
46	0	0	0	0	0
47	0	0	0	0	0
48	0	0	0	0	0
49	0	0	0	0	0
50	0	0	0	0	0

Организация трансформера KeyBERT извлекает униграммы, биграммы и триграммы, однако, этого оказалось недостаточно. Максимальное среднее совпадений составило всего 0,08 слов на один текст.

Таблица 30 – Количество совпадений извлеченных ключевых выражений в художественном подкорпусе, проанализированном экспертами и алгоритмом Topia

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	2	0	1	0	0
2	0	0	0	0	0
3	1	0	0	0	0
4	1	0	0	0	0
5	1	0	0	0	0
6	0	1	0	0	0
7	1	0	0	0	0
8	0	0	0	0	0
9	2	1	0	0	0
10	0	0	1	0	0
11	1	0	0	0	0
12	1	2	1	0	0
13	0	0	0	0	0
14	0	2	0	0	0
15	1	1	0	0	0
16	1	1	0	0	0

Продолжение таблицы 30

17	0	0	0	0	0
18	1	0	0	1	0
19	0	0	0	0	0
20	1	1	0	0	0
21	0	0	0	0	0
22	1	0	0	0	0
23	0	1	0	0	0
24	0	0	0	0	0
25	0	1	0	0	0
26	0	1	1	1	0
27	0	0	0	0	0
28	0	0	0	0	0
29	0	0	0	0	0
30	0	0	0	0	0
31	0	1	0	0	0
32	0	0	0	0	0
33	0	0	0	0	0
34	1	0	0	0	0
35	0	0	0	0	0
36	0	0	0	0	0
37	0	1	0	0	0
38	0	0	0	0	0
39	0	0	0	0	0
40	0	0	0	0	0
41	0	0	0	0	0
42	2	2	0	0	0
43	1	1	0	0	0
44	0	0	0	0	0
45	0	0	0	0	0
46	0	0	0	0	0
47	0	1	0	0	0
48	1	0	0	0	0
49	0	0	0	0	0
50	0	0	0	0	0

Ключевые выражения алгоритма Topia, при их небольшом количестве на выходе, имеют удовлетворительный показатель максимального среднего совпадений – 0,4 слова на один текст. При условии того, что алгоритм построен для извлечения терминов из текстов, ключевые выражения вполне отражают суть повествования художественного текста каждой обработанной главы. Среди

выражений присутствуют как имена действующих лиц, так и существительные, обозначающие предметы взаимодействия с персонажами или ситуации, в которых указанные персонажи попадают. Алгоритм довольно хорошо справляется с большим художественным текстом глав избранных произведений.

Таблица 31 – Количество совпадений извлеченных ключевых выражений в художественном подкорпусе, проанализированном экспертами и алгоритмом Yake

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	2	0	1	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	1	1	1	0	0
5	1	0	0	0	0
6	1	0	0	0	0
7	1	0	0	0	0
8	0	0	0	0	0
9	1	1	0	0	0
10	1	0	0	0	0
11	1	0	0	0	0
12	1	2	1	0	0
13	1	1	0	0	0
14	1	2	0	0	0
15	1	0	0	1	1
16	1	1	0	0	1
17	1	1	1	1	1
18	1	1	0	1	0
19	1	1	0	0	1
20	1	1	0	0	1
21	0	1	0	0	0
22	1	0	0	0	0
23	0	1	0	0	0
24	0	0	0	0	0
25	0	1	0	0	0
26	0	0	1	1	0
27	0	0	0	0	0
28	0	0	0	0	0
29	0	0	0	0	0
30	0	0	0	0	0
31	0	0	0	0	0
32	0	0	0	0	0

Продолжение таблицы 31

33	0	0	0	0	0
34	0	1	0	0	0
35	0	1	0	0	0
36	0	0	0	0	0
37	0	1	0	0	0
38	0	0	0	0	0
39	0	0	0	0	0
40	0	1	0	0	0
41	0	0	0	0	0
42	2	2	0	0	0
43	0	1	0	0	0
44	0	1	1	0	1
45	0	0	0	0	0
46	0	0	0	0	0
47	0	0	0	0	0
48	0	0	0	0	0
49	0	0	0	0	0
50	0	0	0	0	0

Алгоритм Yake имеет схожее с предыдущим алгоритмом количество совпадений, среднее совпадений равно 0,46 слова на текст. Выдача алгоритма состоит из ключевых выражений разных частей речи, что является положительным фактом при проведении процедуры сравнения данных выражений с разметкой экспертов. Yake извлекает имена собственные, что так же часто встречается в экспертных решениях.

Таблица 32 – Количество совпадений извлеченных ключевых выражений в художественном подкорпусе, проанализированном экспертами и алгоритмом TextRank

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	2	0	1	0	0
2	0	0	0	0	0
3	1	0	0	0	0
4	1	0	0	0	0
5	1	0	0	0	0
6	0	0	0	0	0
7	1	0	0	0	0
8	0	0	0	0	0
9	2	1	0	0	0

Продолжение таблицы 32

10	0	0	1	0	0
11	2	0	0	0	0
12	1	2	0	0	0
13	0	0	0	0	0
14	1	2	0	0	0
15	1	1	0	0	0
16	1	1	0	0	0
17	0	0	0	0	0
18	0	0	0	0	0
19	0	0	0	0	0
20	1	1	0	0	0
21	0	1	0	0	0
22	1	0	0	0	0
23	0	1	1	0	0
24	0	0	0	0	0
25	0	0	0	0	0
26	0	0	0	0	0
27	0	0	0	0	0
28	0	0	0	0	0
29	0	0	0	0	0
30	0	0	0	0	0
31	0	0	0	0	0
32	0	0	0	0	0
33	0	0	0	0	0
34	1	0	0	0	0
35	1	0	1	0	0
36	0	0	0	0	0
37	0	1	0	0	0
38	0	0	0	0	0
39	0	0	0	0	0
40	1	1	1	0	0
41	0	0	0	0	0
42	1	1	0	0	0
43	0	0	0	0	0
44	0	1	1	0	0
45	0	0	0	0	0
46	0	0	0	0	0
47	0	1	0	0	0
48	1	0	0	0	0
49	0	0	0	0	0
50	0	0	0	0	0

Ключевые выражения экстрактора TextRank состоят из разных частей речи и имен собственных. Максимальное среднее совпадений алгоритма и экспертов составляет 0,42 слов на текст. Недостатком алгоритма является многократное повторение уже извлеченного выражения, однако, это не помешало иметь достаточно высокий показатель среднего совпадений.

Таблица 33 – Количество совпадений извлеченных ключевых выражений в художественном подкорпусе, проанализированном экспертами и алгоритмом tf-idf

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	2	0	0	0	0
2	0	0	0	0	0
3	1	0	0	0	0
4	1	0	0	0	0
5	1	0	0	0	0
6	1	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	1	1	0	0	0
10	0	0	1	0	0
11	1	0	0	0	0
12	1	2	0	0	0
13	0	0	0	0	0
14	1	2	0	0	0
15	0	1	0	0	0
16	1	1	0	0	0
17	0	0	0	0	0
18	1	0	0	1	0
19	0	0	0	0	0
20	1	1	0	0	0
21	0	1	0	0	0
22	1	0	0	0	0
23	0	0	0	0	0
24	0	0	0	0	0
25	0	0	0	0	0
26	0	1	0	0	0
27	0	0	0	0	0
28	0	0	0	0	0
29	0	0	0	0	0
30	0	0	0	0	0

Продолжение таблицы 33

31	0	0	0	0	0
32	0	0	0	0	0
33	0	0	0	0	0
34	1	0	0	0	0
35	0	0	0	0	0
36	0	0	0	0	0
37	0	1	0	0	0
38	0	0	0	0	0
39	0	0	0	0	0
40	1	2	1	0	0
41	0	0	0	0	0
42	2	2	0	0	0
43	0	0	0	0	0
44	0	1	1	0	0
45	0	0	0	0	0
46	0	0	0	0	0
47	0	1	0	0	0
48	1	0	0	0	0
49	0	0	0	0	0
50	0	0	0	0	0

Алгоритм tf-idf организует выдачу ключевых выражений, состоящих так же из разных частей речи, кроме глаголов. Этот факт уменьшает количество совпадений с экспертным решением, таким образом, максимальное среднее составляет 0,38 слова на текст. Алгоритм извлекает имена собственные, поэтому действующие лица, определенные экспертами, как ключевые выражения, нашли совпадение с выдачей алгоритма.

Таблица 34 – Количество совпадений извлеченных ключевых выражений в художественном подкорпусе, проанализированном экспертами и алгоритмом Chi-square

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	1	0	0	0	0
2	0	0	0	0	0
3	1	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0

Продолжение таблицы 34

8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	1	0	0
11	0	0	0	0	0
12	0	0	0	0	0
13	0	0	0	0	0
14	0	1	0	0	0
15	0	0	0	0	0
16	0	1	1	1	1
17	0	0	0	0	0
18	0	0	0	0	0
19	0	0	0	0	0
20	0	1	1	0	0
21	0	0	0	0	0
22	0	0	0	0	0
23	0	0	0	0	0
24	0	0	0	0	0
25	0	0	0	0	0
26	0	0	1	0	0
27	0	0	0	0	0
28	0	0	0	0	0
29	0	0	0	0	0
30	0	0	0	0	0
31	1	0	0	0	0
32	1	0	0	0	0
33	0	0	0	0	0
34	1	0	0	0	0
35	0	0	0	0	0
36	0	0	0	0	0
37	2	0	1	0	0
38	1	0	1	1	1
39	0	1	0	1	0
40	1	1	1	1	0
41	0	0	0	0	0
42	0	0	0	0	0
43	0	0	0	0	0
44	0	0	0	0	0
45	0	1	1	0	0
46	0	0	0	0	0
47	0	0	0	0	0
48	0	0	0	0	0

Продолжение таблицы 34

49	0	0	0	0	0
50	0	0	0	0	0

Ключевые выражения метрики Chi-square состоят из биграмм-кандидатов на коллокации, что уже является фактом, препятствующим появлению большого числа совпадений. Максимальное среднее совпадений алгоритма и экспертов составило 0,18 слов на текст. Среди коллокаций присутствует большое множество сюжетных событий, описаний, персонажей и действий.

Таблица 35 – Количество совпадений извлеченных ключевых выражений в художественном подкорпусе, проанализированном экспертами и алгоритмом PullEnti

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	2	0	2	0	0
2	2	0	1	0	0
3	2	0	0	0	0
4	1	1	0	0	0
5	3	0	1	2	0
6	2	2	0	1	0
7	2	0	1	0	0
8	1	0	1	0	0
9	1	0	0	0	0
10	2	1	2	0	0
11	3	0	0	0	0
12	2	1	1	0	0
13	2	1	1	0	1
14	2	1	0	0	0
15	1	0	0	0	0
16	1	2	2	0	1
17	2	2	2	1	1
18	2	1	1	1	1
19	2	0	3	0	0
20	1	2	3	1	0
21	2	1	1	1	1
22	1	0	0	1	0
23	2	0	1	0	0
24	1	0	1	0	0
25	1	0	1	0	0
26	1	1	2	1	0

Продолжение таблицы 35

27	1	1	1	1	1
28	2	0	1	0	0
29	1	0	0	1	0
30	2	0	0	0	0
31	2	1	0	0	0
32	3	1	2	1	1
33	3	1	1	0	0
34	3	1	1	0	0
35	4	3	1	0	0
36	4	1	0	0	0
37	3	2	2	0	0
38	0	1	0	0	0
39	1	2	1	1	0
40	1	2	3	1	1
41	2	1	0	1	0
42	3	2	2	0	0
43	3	2	1	0	1
44	1	1	1	0	1
45	1	1	1	1	1
46	1	1	1	1	1
47	0	1	0	0	0
48	3	2	0	1	1
49	1	0	0	0	0
50	2	0	0	1	0

Сервис PullEnti предоставляет огромный выбор ключевых сущностей, как они называются в самом интерфейсе демо-версии программы. Среди них есть существительные, глаголы, имена собственные, прилагательные. Служебные слова и местоимения в выдачи не прошли. Большой текст статьи обеспечил большую выдачу ключевых выражений с самыми разнообразными ситуациями – описания действий, сами действия, характеристики персонажей, слова из диалогов и так далее. Максимальное среднее совпадений составило 1,82 слова на текст.

Таблица 36 – Количество совпадений извлеченных ключевых выражений в художественном подкорпусе, проанализированном экспертами и алгоритмом Log-likelihood

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	1	0	1	0	0
2	0	0	1	0	0
3	1	0	0	0	0
4	1	0	0	0	0
5	0	0	0	0	0
6	1	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0
11	1	0	1	0	0
12	1	1	1	0	0
13	0	0	0	0	0
14	0	2	0	0	0
15	0	0	0	0	0
16	0	0	0	0	0
17	0	0	0	0	0
18	2	1	0	1	1
19	1	0	0	0	0
20	1	2	1	0	0
21	0	0	0	0	0
22	1	1	0	0	0
23	1	0	1	0	0
24	0	0	0	0	0
25	0	0	0	0	0
26	0	0	0	0	0
27	0	0	0	0	0
28	0	0	0	0	0
29	0	0	0	0	0
30	0	0	0	0	0
31	0	1	0	0	0
32	1	0	0	0	0
33	0	0	0	0	0
34	2	2	1	1	0
35	0	1	0	0	0
36	0	0	0	0	0
37	0	0	0	0	0
38	0	0	0	0	0
39	0	0	0	0	0
40	1	2	1	1	0
41	0	0	0	1	0
42	1	1	0	1	1

Продолжение таблицы 36

43	0	1	0	0	0
44	0	1	1	0	1
45	0	1	1	0	0
46	0	0	0	0	0
47	0	0	0	0	0
48	1	1	0	0	0
49	0	0	0	0	0
50	0	0	0	0	0

Коллокации алгоритма Log-likelihood являются строго биграммами, что автоматически уменьшило количество совпадений, исключив появление униграмм у алгоритма и экспертов. Однако, максимальное среднее совпадений составило 0,36 слов на текст. Данный алгоритм получает схожие результаты с остальными алгоритмами.

Таблица 37 демонстрирует полученные результаты совпадений ключевых выражений, извлеченных автоматически и вручную. В ней содержатся значения суммы и среднего арифметического совпадений. Максимальные показатели получились у эксперта номер 1 и экстрактора PullEnti, сумма совпадений равна 91, среднее – 1,82. Минимальные показатели у эксперта номер 5 и алгоритмов TF-IDF, TextRank, KeyBERT, и Topia, сумма и среднее равны 0. Значения остальных экспертов и алгоритмов находятся в пределах максимального и минимального значений.

Рисунок 4 показывает диаграмму результатами процедуры совпадений ключевых выражений, извлеченных алгоритмами и размеченных экспертами.

Таблица 37 – Результаты оценки совпадений работы алгоритмов и экспертного решения в художественном подкорпусе

Алгоритм		ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
Log-likelihood	сумма	18	18	10	5	3
	среднее					
Хи-квадрат	сумма	9	6	8	4	2
	среднее	0,18	0,12	0,16	0,08	0,04
TF-IDF	сумма	19	17	3	1	0
	среднее	0,38	0,34	0,06	0,02	0

Продолжение таблицы 37

PullEnti	сумма	91	43	46	19	13
	среднее	1,82	0,86	0,92	0,38	0,26
YAKE	сумма	21	23	6	4	6
	среднее	0,42	0,46	0,12	0,08	0,12
RAKE	сумма	76	41	36	17	6
	среднее	1,52	0,82	0,72	0,34	0,12
TextRank	сумма	21	15	6	0	0
	среднее	0,42	0,3	0,12	0	0
KeyBERT	сумма	4	4	2	4	0
	среднее	0,08	0,08	0,04	0,08	0
Topia	сумма	20	18	4	2	0
	среднее	0,4	0,36	0,08	0,04	0

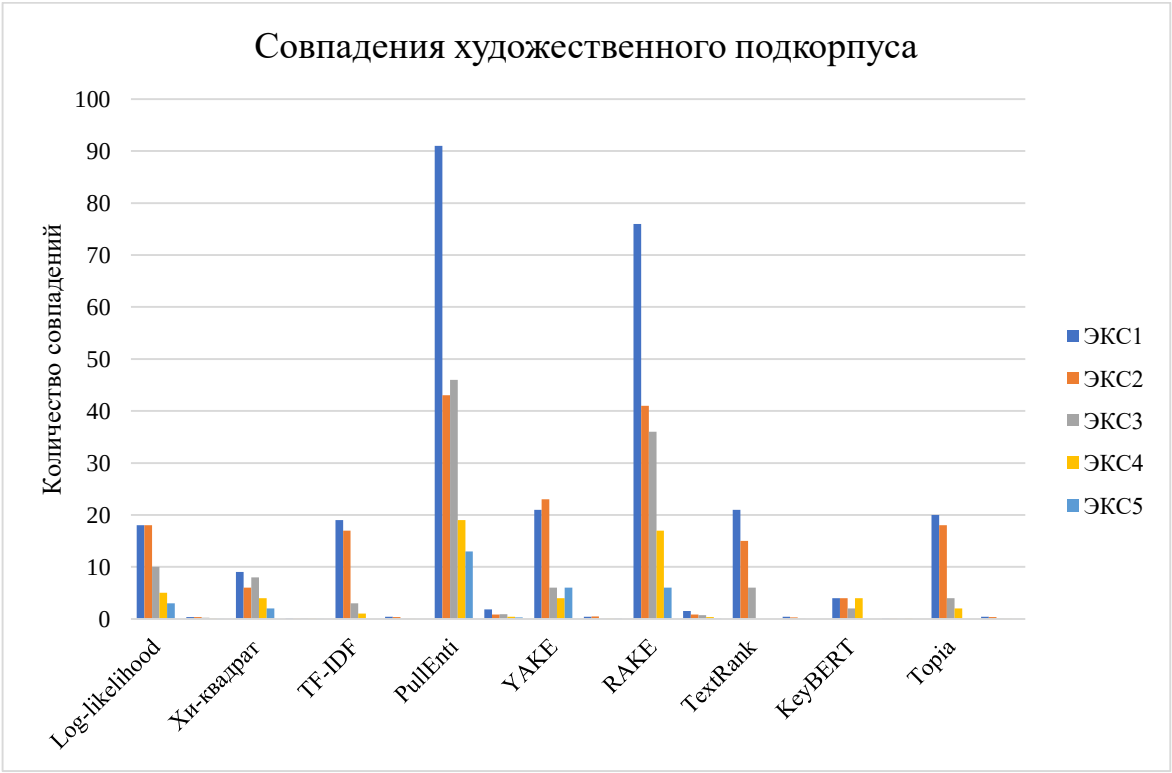

Рисунок 4. Диаграмма совпадений художественного подкорпуса

6. Оценка результатов экспериментов

6.1. Теоретические основания оценки результатов

Важной задачей является оценка эффективности результатов используемого метода. Эффективность оценивается релевантностью автоматически найденных ключевых выражений в документе по отношению к заведомо известным ключевым выражениям.

Традиционно для оценки эффективности используются такие показатели, как точность, полнота и f-мера. В данном исследовании эти показатели не способны в полной мере отразить суть полученных результатов и оценить их в отличие от оценки доли совпадений, которая была приведена в прошлой главе. Однако было принято решение провести традиционную оценку результатов для того, чтобы проверить эффективность каждого алгоритма каждого подкорпуса относительно эталонной выборки, т.е. разметки экспертов.

Для оценки совпадений ключевых выражений, извлеченных алгоритмами и размеченных экспертами, за основу была взята статья исследователей, занимающихся созданием веб-корпуса тюркских языков [31]. В данном исследовании точность и полнота рассматриваются как метрики оценки правильно расположенных словоформ.

Решение экспертов расценивается как золотой стандарт и считается релевантными словами для всех текстов всех подкорпусов.

Таким образом, согласно определнию, точность (Precision) – это отношение числа релевантных ключевых выражений, найденных автоматически, к общему числу найденных ключевых выражений в документе. Формула для данного исследования будет выглядеть следующим образом (6):

$$P = \frac{H}{H + I}, \qquad (6)$$

где P – точность, H – множество совпадений, I – множество ключевых выражений, выданных алгоритмом.

Определение полноты (Recall) звучит как отношение числа релевантных ключевых выражений, найденных автоматически, к общему числу релевантных ключевых выражений в документе (7).

$$R = \frac{H}{H + D}, \qquad (7)$$

где R – полнота, H – множество совпадений, D – множество ключевых выражений, выделенных экспертами.

F-мера (F-score, F-measure) – объединение точности и полноты в одной усредненной величине, определяется как взвешенное гармоническое среднее точности и полноты (8).

$$F = 2 * \frac{P * R}{P + R}, \qquad (8)$$

где F – мера, P – точность, R – полнота.

6.2. Проведение расчетов оценки результатов

Расчеты точности для каждого алгоритма проходили с учетом количества выражений, выданных алгоритмом. Расчеты полноты проводились с учетом количества ключевых выражений золотого стандарта, то есть выражений, размеченных экспертами. В данном эксперименте положительным результатом считается совпадение хотя бы одного ключевого выражения в результатах алгоритмов и экспертов. Таким образом, в силу сформулированной гипотезы и экспериментального дизайна исследования, показатели точности, полноты и f-меры будут сосредоточены в начале интервала [0,1], то есть тяготеть к 0.

Используя формулы из предыдущего параграфа, мы получили следующие результаты точности, полноты и f-меры для каждого алгоритма соответственно.

Точность для алгоритма Log-likelihood публицистического подкорпуса (9):

$$P = \frac{138}{138 + 4162} = 0{,}032093023 \quad (9)$$

Полнота для алгоритма Log-likelihood публицистического подкорпуса (10):

$$R = \frac{138}{138 + 697} = 0{,}165269461 \quad (10)$$

F-мера для алгоритма Log-likelihood публицистического подкорпуса (11):

$$F = 2 * \frac{0{,}032093023 * 0{,}165269461}{0{,}032093023 + 0{,}165269461} = 0{,}053748783 \quad (11)$$

Точность для алгоритма Log-likelihood научного подкорпуса (12):

$$P = \frac{194}{194 + 5000} = 0{,}037350789 \quad (12)$$

Полнота для алгоритма Log-likelihood научного подкорпуса (13):

$$R = \frac{194}{194 + 680} = 0{,}221967963 \quad (13)$$

F-мера для алгоритма Log-likelihood научного подкорпуса (14):

$$F = 2 * \frac{0{,}037350789 * 0{,}221967963}{0{,}037350789 + 0{,}221967963} = 0{,}063941991 \quad (14)$$

Точность для алгоритма Log-likelihood художественного подкорпуса (15):

$$P = \frac{54}{54 + 5000} = 0{,}010684606 \quad (15)$$

Полнота для алгоритма Log-likelihood художественного подкорпуса (16):

$$R = \frac{54}{54 + 666} = 0{,}075 \quad (16)$$

F-мера для алгоритма Log-likelihood художественного подкорпуса (17):

$$F = 2 * \frac{0{,}010684606 * 0{,}075}{0{,}010684606 + 0{,}075} = 0{,}018704538 \quad (17)$$

Алгоритм Log-likelihood для всех трёх подкорпусов в сравнении с экспертным решением справляется почти одинаково для публицистического и

научного подкорпусов: 5,4% и 6,4% соответственно, а для художественного – 1,8%. Данные показатели f-меры говорят об очень низкой эффективности алгоритма относительно экспертного решения.

Точность для алгоритма Chi-square публицистического подкорпуса (18):

$$P = \frac{77}{77 + 4162} = 0{,}018164661 \quad (18)$$

Полнота для алгоритма Chi-square публицистического подкорпуса (19):

$$R = \frac{77}{77 + 697} = 0{,}099483204 \quad (19)$$

F-мера для алгоритма Chi-square публицистического подкорпуса (20):

$$F = 2 * \frac{0{,}018164661 * 0{,}099483204}{0{,}018164661 + 0{,}099483204} = 0{,}030720128 \quad (20)$$

Точность для алгоритма Chi-square научного подкорпуса (21):

$$P = \frac{33}{33 + 5000} = 0{,}006556726 \quad (21)$$

Полнота для алгоритма Chi-square научного подкорпуса (22):

$$R = \frac{33}{33 + 680} = 0{,}04628331 \quad (22)$$

F-мера для алгоритма Chi-square научного подкорпуса (23):

$$F = 2 * \frac{0{,}006556726 * 0{,}04628331}{0{,}006556726 + 0{,}04628331} = 0{,}011486251 \quad (23)$$

Точность для алгоритма Chi-square художественного подкорпуса (24):

$$P = \frac{29}{29 + 5000} = 0{,}005766554 \quad (24)$$

Полнота для алгоритма Chi-square художественного подкорпуса (25):

$$R = \frac{29}{29 + 666} = 0{,}041726619 \quad (25)$$

F-мера для алгоритма Chi-square художественного подкорпуса (26):

$$F = 2 * \frac{0{,}005766554 * 0{,}041726619}{0{,}005766554 + 0{,}041726619} = 0{,}010132774 \quad (26)$$

Результаты оценки алгоритма Chi-square отличаются от предыдущего алгоритма цифрами. Для публицистического подкорпуса f-мера составила 3%,

для научного и художественного – 1%. Результаты говорят о низкой эффективности алгоритма относительно экспертного решения.

Точность для алгоритма TF-IDF публицистического подкорпуса (27):

$$P = \frac{156}{156 + 500} = 0{,}237804878 \quad (27)$$

Полнота для алгоритма TF-IDF публицистического подкорпуса (28):

$$R = \frac{156}{156 + 697} = 0{,}182883939 \quad (28)$$

F-мера для алгоритма TF-IDF публицистического подкорпуса (29):

$$F = 2 * \frac{0{,}237804878 * 0{,}182883939}{0{,}237804878 + 0{,}182883939} = 0{,}206759443 \quad (29)$$

Точность для алгоритма TF-IDF научного подкорпуса (30):

$$P = \frac{99}{99 + 500} = 0{,}165275459 \quad (30)$$

Полнота для алгоритма TF-IDF научного подкорпуса (31):

$$R = \frac{99}{99 + 680} = 0{,}127086008 \quad (31)$$

F-мера для алгоритма TF-IDF научного подкорпуса (32):

$$F = 2 * \frac{0{,}165275459 * 0{,}127086008}{0{,}165275459 + 0{,}127086008} = 0{,}143686502 \quad (32)$$

Точность для алгоритма TF-IDF художественного подкорпуса (33):

$$P = \frac{40}{40 + 500} = 0{,}074074074 \quad (33)$$

Полнота для алгоритма TF-IDF художественного подкорпуса (34):

$$R = \frac{40}{40 + 666} = 0{,}056657224 \quad (34)$$

F-мера для алгоритма TF-IDF художественного подкорпуса (35):

$$F = 2 * \frac{0{,}074074074 * 0{,}056657224}{0{,}074074074 + 0{,}056657224} = 0{,}064205457 \quad (35)$$

Алгоритм TF-IDF имеет достаточно высокие результаты оценки эффективности. Для публицистического подкорпуса TF-IDF имеет самый

большой процент эффективности относительно других алгоритмов – 20,7%. Для научного подкорпуса f-мера равна 14%, для художественного – 6%. Разнообразие результатов от подкорпуса к подкорпусу объяснима увеличением объема текстов и сложностью самих текстов. Результаты алгоритма TF-IDF говорят о достаточно высокой эффективности алгоритма относительно других результатов и экспертного решения.

Точность для алгоритма PullEnti публицистического подкорпуса (36):

$$P = \frac{240}{240 + 3721} = 0{,}06059076 \quad (36)$$

Полнота для алгоритма PullEnti публицистического подкорпуса (37):

$$R = \frac{240}{240 + 697} = 0{,}256136606 \quad (37)$$

F-мера для алгоритма PullEnti публицистического подкорпуса (38):

$$F = 2 * \frac{0{,}06059076 * 0{,}256136606}{0{,}06059076 + 0{,}256136606} = 0{,}097999183 \ (38)$$

Точность для алгоритма PullEnti научного подкорпуса (39):

$$P = \frac{281}{281 + 25130} = 0{,}011058203 \quad (39)$$

Полнота для алгоритма PullEnti научного подкорпуса (40):

$$R = \frac{281}{281 + 680} = 0{,}292403746 \quad (40)$$

F-мера для алгоритма PullEnti научного подкорпуса (41):

$$F = 2 * \frac{0{,}011058203 * 0{,}292403746}{0{,}011058203 + 0{,}292403746} = 0{,}021310481 \ (41)$$

Точность для алгоритма PullEnti художественного подкорпуса (42):

$$P = \frac{212}{212 + 43205} = 0{,}00488288 \quad (42)$$

Полнота для алгоритма PullEnti художественного подкорпуса (43):

$$R = \frac{212}{212 + 666} = 0{,}241457859 \quad (43)$$

F-мера для алгоритма PullEnti художественного подкорпуса (44):

$$F = 2 * \frac{0{,}00488288 * 0{,}241457859}{0{,}00488288 + 0{,}241457859} = 0{,}009572186 \quad (44)$$

Оценка результатов сервиса PullEnti получилась средней и низкой относительного результатов других экстракторов. Для публицистического подкорпуса – 9,8%, для научного – 2%, для художественного – 1%. Несмотря на самые высокие результаты совпадений ключевых выражений экстрактора и экспертов, оценка эффективности получилась весьма небольшой. Причиной таких маленьких показателей является большое количество выдачи ключевых выражений. По результатам оценки эффективности алгоритма PullEnti может быть признан низко эффективным.

Точность для алгоритма YAKE публицистического подкорпуса (45):

$$P = \frac{99}{99 + 1000} = 0{,}090081893 \quad (45)$$

Полнота для алгоритма YAKE публицистического подкорпуса (46):

$$R = \frac{99}{99 + 697} = 0{,}124371859 \quad (46)$$

F-мера для алгоритма YAKE публицистического подкорпуса (47):

$$F = 2 * \frac{0{,}090081893 * 0{,}124371859}{0{,}090081893 + 0{,}124371859} = 0{,}104485488 \quad (47)$$

Точность для алгоритма YAKE научного подкорпуса (48):

$$P = \frac{120}{120 + 1000} = 0{,}107142857 \quad (48)$$

Полнота для алгоритма YAKE научного подкорпуса (49):

$$R = \frac{120}{120 + 680} = 0{,}15 \quad (49)$$

F-мера для алгоритма YAKE научного подкорпуса (50):

$$F = 2 * \frac{0{,}107142857 * 0{,}15}{0{,}107142857 + 0{,}15} = 0{,}125 \quad (50)$$

Точность для алгоритма YAKE художественного подкорпуса (51):

$$P = \frac{60}{60 + 1000} = 0{,}056603774 \quad (51)$$

Полнота для алгоритма YAKE художественного подкорпуса (52):

$$R = \frac{60}{60 + 666} = 0{,}082644628 \quad (52)$$

F-мера для алгоритма YAKE художественного подкорпуса (53):

$$F = 2 * \frac{0{,}056603774 * 0{,}082644628}{0{,}056603774 + 0{,}082644628} = 0{,}06718925 \quad (53)$$

Результаты оценки эффективности алгоритма YAKE являются средними относительно других алгоритмов. Для публицистического f-мера равна 10%, для научного – 12,5%, для художественного – 6,7%. Данные результаты могут быть интерпретированы, как средние и даже высокие для художественного подкорпуса. Однако, эффективность алгоритма YAKE признается низкой.

Точность для алгоритма RAKE публицистического подкорпуса (54):

$$P = \frac{180}{180 + 3326} = 0{,}051340559 \quad (54)$$

Полнота для алгоритма RAKE публицистического подкорпуса (55):

$$R = \frac{180}{180 + 697} = 0{,}205245154 \quad (55)$$

F-мера для алгоритма RAKE публицистического подкорпуса (56):

$$F = 2 * \frac{0{,}051340559 * 0{,}205245154}{0{,}051340559 + 0{,}205245154} = 0{,}082135524 \quad (56)$$

Точность для алгоритма RAKE научного подкорпуса (57):

$$P = \frac{225}{225 + 23278} = 0{,}009573246 \quad (57)$$

Полнота для алгоритма RAKE научного подкорпуса (58):

$$R = \frac{225}{225 + 680} = 0{,}248618785 \quad (58)$$

F-мера для алгоритма RAKE научного подкорпуса (59):

$$F = 2 * \frac{0{,}009573246 * 0{,}248618785}{0{,}009573246 + 0{,}248618785} = 0{,}018436578 \quad (59)$$

Точность для алгоритма RAKE художественного подкорпуса (60):

$$P = \frac{176}{176 + 50502} = 0{,}003472907 \quad (60)$$

Полнота для алгоритма RAKE художественного подкорпуса (61):

$$R = \frac{176}{176 + 666} = 0{,}209026128 \quad (61)$$

F-мера для алгоритма RAKE художественного подкорпуса (62):

$$F = 2 * \frac{0{,}003472907 * 0{,}209026128}{0{,}003472907 + 0{,}209026128} = 0{,}006945815 \quad (62)$$

Алгоритм RAKE по эффективности имеет очень разные результаты: средние для публицистического подкорпуса f-мера равна 8%, для научного – 1,8%, для художественного – 0,6%. Несмотря на высокие показатели количества совпадений ключевых выражений алгоритма и экспертного решения, результаты эффективности получились достаточно низкими для научного и художественного подкорпусов. Данное положение дел объясняется большим количеством ключевых выражений при выдаче алгоритма.

Точность для алгоритма TextRank публицистического подкорпуса (63):

$$P = \frac{136}{136 + 947} = 0{,}125577101 \quad (63)$$

Полнота для алгоритма TextRank публицистического подкорпуса (64):

$$R = \frac{136}{136 + 697} = 0{,}163265306 \quad (64)$$

F-мера для алгоритма TextRank публицистического подкорпуса (65):

$$F = 2 * \frac{0{,}125577101 * 0{,}163265306}{0{,}125577101 + 0{,}163265306} = 0{,}141962422 \quad (65)$$

Точность для алгоритма TextRank научного подкорпуса (66):

$$P = \frac{79}{79 + 2621} = 0{,}029259259 \quad (66)$$

Полнота для алгоритма TextRank научного подкорпуса (67):

$$R = \frac{79}{79 + 680} = 0{,}104084321 \quad (67)$$

F-мера для алгоритма TextRank научного подкорпуса (68):

$$F = 2 * \frac{0{,}029259259 * 0{,}104084321}{0{,}029259259 + 0{,}104084321} = 0{,}045677942 \quad (68)$$

Точность для алгоритма TextRank художественного подкорпуса (69):

$$P = \frac{42}{42 + 6392} = 0{,}006527821 \quad (69)$$

Полнота для алгоритма TextRank художественного подкорпуса (70):

$$R = \frac{42}{42 + 666} = 0{,}059322034 \quad (70)$$

F-мера для алгоритма TextRank художественного подкорпуса (71):

$$F = 2 * \frac{0{,}006527821 * 0{,}059322034}{0{,}006527821 + 0{,}059322034} = 0{,}011761411 \quad (71)$$

Алгоритм TextRank имеет достаточно высокие результаты оценки эффективности относительно других алгоритмов. Для публицистического f-мера равна 14%, для научного – 4,5%, для художественного – 1%. Градиентное уменьшение f-меры для всех трех подкорпусов объясняется увеличением объемов текстов и сложности. Эффективность алгоритма TextRank относительно других результатов получается средней.

Точность для алгоритма KeyBERT публицистического подкорпуса (72):

$$P = \frac{35}{35 + 750} = 0{,}044585987 \quad (72)$$

Полнота для алгоритма KeyBERT публицистического подкорпуса (73):

$$R = \frac{35}{35 + 697} = 0{,}047814208 \quad (73)$$

F-мера для алгоритма KeyBERT публицистического подкорпуса (74):

$$F = 2 * \frac{0{,}044585987 * 0{,}047814208}{0{,}044585987 + 0{,}047814208} = 0{,}046143705 \quad (74)$$

Точность для алгоритма KeyBERT научного подкорпуса (75):

$$P = \frac{13}{13 + 750} = 0{,}017038008 \quad (75)$$

Полнота для алгоритма KeyBERT научного подкорпуса (76):

$$R = \frac{13}{13 + 680} = 0{,}018759019 \quad (76)$$

F-мера для алгоритма KeyBERT научного подкорпуса (77):

$$F = 2 * \frac{0{,}017038008 * 0{,}018759019}{0{,}017038008 + 0{,}018759019} = 0{,}017857143 \quad (77)$$

Точность для алгоритма KeyBERT художественного подкорпуса (78):

$$P = \frac{14}{14 + 750} = 0{,}018324607 \quad (78)$$

Полнота для алгоритма KeyBERT художественного подкорпуса (79):

$$R = \frac{14}{14 + 666} = 0{,}020588235 \quad (79)$$

F-мера для алгоритма KeyBERT художественного подкорпуса (80):

$$F = 2 * \frac{0{,}018324607 * 0{,}020588235}{0{,}018324607 + 0{,}020588235} = 0{,}019390582 \quad (80)$$

Оценка эффективности алгоритма KeyBERT получила низкие показатели по всем трем подкорпусам. Так, например, f-мера для публицистического подкорпуса равна 4,6%, для научного – 1,8% и для художественного – 1,9%. Такие результаты объясняются спецификой выдачи самого трансформера и малым количеством совпадений с результатами экспертов. Эффективность алгоритма KeyBERT признается низкой.

Точность для алгоритма Topia публицистического подкорпуса (81):

$$P = \frac{121}{121 + 500} = 0{,}194847021 \quad (81)$$

Полнота для алгоритма Topia публицистического подкорпуса (82):

$$R = \frac{121}{121 + 697} = 0{,}14792176 \quad (82)$$

F-мера для алгоритма Topia публицистического подкорпуса (83):

$$F = 2 * \frac{0{,}194847021 * 0{,}14792176}{0{,}194847021 + 0{,}14792176} = 0{,}168172342 \quad (83)$$

Точность для алгоритма Topia научного подкорпуса (84):

$$P = \frac{115}{115 + 500} = 0{,}18699187 \quad (84)$$

Полнота для алгоритма Topia научного подкорпуса (85):

$$R = \frac{115}{115 + 680} = 0{,}144654088 \quad (85)$$

F-мера для алгоритма Topia научного подкорпуса (86):

$$F = 2 * \frac{0{,}18699187 * 0{,}144654088}{0{,}18699187 + 0{,}144654088} = 0{,}163120567 \quad (86)$$

Точность для алгоритма Topia художественного подкорпуса (87):

$$P = \frac{44}{44 + 500} = 0{,}080882353 \quad (87)$$

Полнота для алгоритма Topia художественного подкорпуса (88):

$$R = \frac{44}{44 + 666} = 0{,}061971831 \quad (88)$$

F-мера для алгоритма Topia художественного подкорпуса (89):

$$F = 2 * \frac{0{,}080882353 * 0{,}061971831}{0{,}080882353 + 0{,}061971831} = 0{,}070175439 \quad (89)$$

Результаты оценки эффективности алгоритма Topia имеют самые большие значения для научного и художественного подкорпусов. Таким образом, f-мера для публицистического подкорпуса составляет 16,8%, для научного и художественного – 7%. Такие высокие оценки объяснимы небольшим количеством ключевых выражений на выходе алгоритма, а также достаточно средним количеством совпадений с результатами экспертов. Эффективность алгоритма Topia признается высокой относительно других алгоритмов, участвовавших в исследовании.

По результатам проведенных расчетов оценки эффективности можно однозначно утвердить, что ключевые выражения зависят от местоположения в тексте. В настоящем исследовании доказано, что ключевые выражения находятся в начале текстов разных функциональных стилей (заголовки новостей, аннотации научных статей, первые абзацы глав художественных текстов). Автоматическое извлечение ключевых выражений имеет использовалось для проверки гипотезы о существовании связи местоположения и ключевого выражения. Малое количество совпадений, небольшое среднее совпадений на текст и невысокая оценка f-меры подтверждают гипотезу о наличии ключевых выражений в начале текста.

Глава 7. Разработка собственного экстрактора ключевых выражений

7.1. Исследование структуры ключевого выражения на подкорпусе научных текстов

Для задачи извлечения ключевых выражений из текстов необходимо понять, что есть ключевое выражение в терминах формализованного языка. Таким образом, было проведено исследование структуры ключевых выражений с использованием подкорпуса научных текстов, в которых авторы по собственному усмотрению выделили ключевые выражения.

В отечественной литературе были предприняты попытки исследования морфологических шаблонов для автоматического извлечения терминологии [1]. В исследовании принимают участие шаблоны только с биграммными терминами, а также описание анализа избранных авторами шаблонов отсутствует.

В настоящем исследовании структур ключевых выражений все процессы, связанные со сбором ключевых выражений из статей, происходили автоматически. Листинг программы представлен в приложении Б. Получившиеся результаты представлены в таблице 38.

Таблица 38 – Характеристики собранных ключевых выражений

Характеристика	Количество, шт.
Общие характеристики	
Всего статей	120
Всего ключевых выражений из статей	592
Характеристики n-грамм	
Униграммы	195
Биграммы	320
Триграммы	65

Продолжение таблицы 38

Остаток (4 слова и более)	12

Таким образом, преобладающей n-граммой для ключевых выражений является биграмма. Далее целью исследования являлось определить морфологические характеристики каждой n-граммы. Для этого все ключевые выражения были автоматически разделены по количеству слов в строке в разные файлы, а затем был использован частеречный анализатор с заданными морфологическими шаблонами слов и словосочетаний. Листинг программы приведен в приложении В.

Полученные данные позволили создать морфологические модели ключевых выражений, которые приведены в таблице 39 с примерами.

Таблица 39 – Морфологические модели ключевых выражений для n-грамм

Модель	Пример	Количество, шт.
Униграммы		195
[Nnomsing]	*корпус* *словарь*	154
[Nnompl]	*предлоги* *словари*	25
[Adjnomsing]	*Белорусский* *учебный*	3
Названия	*LDA, PyMorphy2*	16
Биграммы		320
[Adjnomsing] + [Nnomsing]	*равномерное распределение* *русский язык*	185
[Adjnompl] + [Nnompl]	*частотные словари* *новые иероглифы*	66
[Nnomsing] + [Ngenpl]	*корпус учебников* *размер конструкций*	26
[Nnompl] + [Ngenpl]	*движения плеч* *меры ассоциации*	9
[Nnomsing] + [Ngensing]	*применимость инструкции* *анализ предложения*	20
[Nnompl] + [Ngensing]	*критерии сопоставимости* *части речи*	9

Продолжение таблицы 39

Имена	*Лев Щерба* *Григорий Винокур*	2
Биграммы с иностранным словом	*wacky технология* *DKPro Similarity*	3
Триграммы		65
[Adjnomsing] + [Adjnomsing] + [Nnomsing]	*Русский ассоциативный словарь* *русский жестовый язык*	13
[Adjnompl] + [Nnompl] + [Ngenpl]	*сопоставимые корпуса текстов* *параллельные корпуса текстов*	8
[Nnomsing] + [Adjgensing] + [Ngensing]	*корпус повседневной речи* *корпус устной речи*	8
[Nnompl] + [Adjgensing] + [Ngensing]	*сценарии гуманитарного образования* *модели частотного поведения*	4
[Nnomsing] + [Adjgenpl] + [Ngenpl]	*корпус ученических текстов* *хранение корпусных данных*	10
[Adjnomsing] + [Nnomsing] + [Ngenpl]	*автоматическое исправление ошибок* *автоматический анализ текстов*	4
[Adjnomsing] + [Nnomsing] + [Ngensing]	*автоматическое распознавание речи* *ручная разметка текста*	5
[Adjnompl] + [Nnompl] + [Ngensing]	*сопоставимые корпуса текстов* *параллельные корпуса текстов*	8
[Adjnompl] + [Adjnompl] + [Nnompl]	*средневековые славянские рукописи* *ижорские народные песни*	5
Триграммы с предлогом	*поиск в корпусе* *предложения с* 𝒳𝒻	2
Триграммы с иностранным словом	*FieldWorks Language Explorer* *язык программирования Python*	3

Все три n-граммы имеют некоторые ключевые выражений, для которых не создан шаблон. Это обусловлено сложностью обобщения всех возможных

вариантов ключевого выражения. Так, например, n-граммы с предлогом требуют отдельного исследования и создания шаблонов с использованием разных предлогов и парадигм склонения употребляющихся с ними частей речи.

По результатам данного исследования можно заключить, что в текстах научных стилей в большинстве случаев употребляются биграммы с использованием существительных и прилагательных в разных парадигмах склонения и в разном числе. Морфологические модели, полученные на данном этапе, будут использоваться в собственном экстракторе по извлечению ключевых выражений.

7.2. Экстрактор ключевых выражений, основанный на грамматике русского языка

7.2.1. Грамматика экстрактора

Алгоритм извлечения ключевых выражений основан на том, что пользователь самостоятельно определяет подходящие под свои нужды грамматические шаблоны для поиска ключевых выражений на основе эмпирических, методологических или теоретических данных. Шаблоны обозначаются в грамматике как последовательность тегов части речи (POS). Поскольку алгоритм извлечения ищет в тексте определенные пользователем последовательности POS, он требует предварительного морфологического анализа. Слова анализируются с помощью морфологического анализатора pymorphy2 [23], с помощью триграмного частеречного алгоритма устраняется неоднозначность, который также и обеспечивает наиболее вероятную последовательность тегов в контексте предложения. Извлеченные ключевые выражения представляют собой n-граммы, которые соответствуют тому, что пользователь определил как допустимую последовательность тегов POS в грамматике. Ключевые выражения ранжируются и сортируются в порядке убывания их оценок.

Грамматика предоставляется извне в виде отдельного текстового файла. Каждая строка в файле должна описывать только один шаблон из n тегов POS (по одному шаблону на строку). Пустые и прокомментированные строки полностью игнорируются (комментарии вводятся с помощью символа «#»), а теги проверяются на соответствие формату, описанному в остальной части этого раздела. Поскольку алгоритм извлечения основан на морфологическом анализе pymorphy2 [23], его номенклатура используется без изменений.

Теги частей речи: 'NOUN' (существительное), 'ADJF' (полное прилагательное), 'ADJS' (краткое прилагательное), 'COMP' (сравнительный), 'VERB' (глагол, личная форма), 'INFN' (глагол, инфинитив), 'PRTF' (полное причастие), 'PRTS' (краткое причастие), 'GRND' (наречие), 'NUMR' (числительное), 'ADVB' (наречие), 'NPRO' (местоимение), 'PRED' (предикатив), 'PREP' (предлог), 'CONJ' (соединение), 'PRCL' (частица), 'INTJ' (междометие).

Теги числа: 'sing' (единственное число), 'plur' (множественное число).

Теги падежей: 'nomn' (именительный падеж), 'gent' (родительный падеж), 'datv' (дательный падеж), 'accs' (винительный падеж), 'ablt' (инструментальный падеж), 'loct' (предложный падеж), 'voct' (звательный падеж), 'gen1' (родительный падеж), 'gen2' (частичный падеж), 'acc2': (второй винительный падеж), 'loc1': (предложный падеж), 'loc2' (местный падеж).

Комбинирование тегов возможно, однако, неграмматические последовательности реже встречаются в текстах. Для классов слов с падежным склонением (существительные, полные прилагательные и причастия, местоимения и числительные) обязательно указывать число и падеж после соответствующей части речи (в указанном порядке), разделенных символом «_».

В таблице 40 приведены все теги грамматических характеристик с расшифровкой на русском языке.

Таблица 40 – Теги грамматических характеристик для ключевых выражений экстрактора

Униграммы	
Существительное единственного числа именительного падежа	NOUN_sing_nomn
Существительное множественного числа именительного падежа	NOUN_plur_nomn
Прилагательное единственного числа именительного падежа	ADJF_sing_nomn
Биграммы	
Прилагательное единственного числа именительного падежа и существительное единственного числа именительного падежа	ADJF_sing_nomn NOUN_sing_nomn
Прилагательное множественного числа именительного падежа и существительное множественного числа именительного падежа	ADJF_plur_nomn NOUN_plur_nomn
Существительное единственного числа именительного падежа и существительное множественного числа родительного падежа	NOUN_sing_nomn NOUN_plur_gent
Существительное множественного числа именительного падежа и существительное множественного числа родительного падежа	NOUN_plur_nomn NOUN_plur_gent
Триграммы	
Прилагательное единственного числа именительного падежа, прилагательное единственного числа именительного падежа и существительное единственного числа именительного падежа	ADJF_sing_nomn ADJF_sing_nomn NOUN_sing_nomn
Прилагательное множественного числа именительного падежа, существительное множественного числа именительного падежа и существительное множественного числа родительного падежа	ADJF_plur_nomn NOUN_plur_nomn NOUN_plur_gent
Существительное единственного числа именительного падежа, прилагательное единственного числа родительного падежа и существительное единственного числа родительного падежа	NOUN_sing_nomn ADJF_sing_gent NOUN_sing_gent

Продолжение таблицы 40

Существительное множественного числа именительного падежа, прилагательное единственного числа родительного падежа и существительное единственного числа родительного падежа	NOUN_plur_nomn ADJF_sing_gent NOUN_sing_gent

В случае если тег в грамматике не следует строго этому формату, он будет считаться недействительным, и код выдаст исключение. На остальные классы слов, не использующие падежные склонения, ссылаются только части речи, например, PREP.

Каждая последовательность тегов должна быть выражена в одной строке в предполагаемом порядке и с тегами, разделенными пробелом. Например, последовательность, состоящая из предлога, за которым следует прилагательное в единственном числе и существительное в единственном числе, оба в локальном падеже, имеет следующий формат: PRED ADJF_sing_loct NOUN_sing_loct.

Примеры последовательностей тегов:

NOUN_sing_nomn – существительное в единственном числе именительном падеже;

ADJF_sing_nomn NOUN_sing_nomn – словосочетание из прилагательного в единственном числе именительном падеже и существительного в единственном числе именительном падеже;

PRED ADJF_sing_loct NOUN_sing_loct NOUN_sing_gent – словосочетание из предлога, прилагательного в единственном числе локальном падеже, существительного в единственном числе локальном падеже и существительного в единственном числе родительном падеже.

7.2.2. Метрика экстрактора

Извлеченные ключевые выражения-кандидаты ранжируются в соответствии с метрикой, которая учитывает длину n-граммы и количество раз,

когда лемматизированные формы слов, составляющих ключевое выражение, появляются в тексте (90):

$$score(k) = \frac{c(l) * len(k)^w}{c(t)}, \qquad (90)$$

где *c(l)* – сумма всех частот в тексте лемматизированных форм слов, составляющих ключевое выражение; *len(k)* – это количество элементов (слов), содержащихся в ключевом выражении; *w* – это значение, которое пользователь передает в качестве аргумента; *c(t)* – количество токенов, содержащихся в тексте.

Для униграммных ключевых выражений значение равно 1. Для ключевых выражений из n слов значение равно n. Аргумент призван повлиять на длину ключевых выражений. Ключевые выражения, содержащие более одного слова, обычно получают более высокие оценки, чем униграммы. Если положительное значение веса меньше 1, это сгладит разницу в оценках между ключевыми выражениями разной длины. Положительный вес с положительным значением больше 1 увеличит оценки ключевых выражений в зависимости от их длины. И наоборот, отрицательный вес будет наказывать ключевые выражения соответственно их длине. Вес 1 или 0 не будет иметь никакого эффекта. Количество токенов – это просто нормализационное значение, поэтому ключевые выражения получают пропорциональные оценки независимо от длины их исходных текстов.

7.2.3. Реализация экстрактора

Обязательным условием работы экстрактора является наличие установленной библиотеки pymorphy2 [23] в среде языка программирования Python3. Процесс устранения неоднозначности после морфологического анализа основан на вероятностях в файле «transition_probabilities.json», поэтому это также является обязательным требованием. Пользователь должен предоставить алгоритму файл, содержащий грамматику, оформленную в соответствии с правилами форматирования, описанными в разделе 7.2.1.

Пример кода:

```
from kw_extractor import POStagger, Keywords #1
tagger = POStagger('trainstion_probabilities.json') #2
keywords = Keywords('rules.txt') #3
filename = 'text.txt' #4
tagged_text, token_count = tagger.parse(filename) #5
keywords.extract(tagged_text, token_count, 0.01, 1) #6,
```

где #1 импортирует классы «POStagger» и «Ключевые выражения» из файла «kw_extractor.py»; #2 объект tagger инициализируется из класса POStagger файлом, который содержит вероятности перехода в качестве аргумента; #3 объект «ключевые выражения» инициализируется из класса «Ключевые выражения» с помощью текстового файла, который содержит грамматику в качестве аргумента; #4 путь к текстовому файлу для обработки; #5 метод «parse» объекта «tagger» принимает текст для обработки в качестве входных данных и возвращает помеченный текст и количество содержащихся в нем токенов; #6 метод «extract» объекта «ключевые выражения» принимает 4 аргумента. Первый аргумент – это текст с тегами, второй – количество токенов в тексте. Наличие этих двух аргументов обязательно. Третий аргумент – это порог, который будет отфильтровывать ключевые выражения с оценками ниже заданного значения. По умолчанию это значение установлено на «0,001». Четвертый и последний аргумент – это вес, который используется для наказания или поощрения оценки ключевых выражений в зависимости от их количества слов. По умолчанию вес установлен на 1 (без эффекта).

7.3. Результаты работы экстрактора для публицистического, научного и художественного подкорпусов текстов

В качестве последовательностей тегов для данной работы были взяты результаты, полученные в параграфе 7.1. В таблице 39 приведены теги, грамматическая структура которых была исследована на основе ключевых

выражений научного подкорпуса настоящей работы. Листинг программы приведен в приложении Г.

В таблицах 41-43 приведены результаты совпадений, извлеченных экстрактором ключевых выражений и результатов работы экспертов публицистического, научного и художественного подкорпусов соответственно.

Таблица 41 – Количество совпадений извлеченных ключевых выражений в публицистическом подкорпусе, проанализированном экспертами экстрактором на основе грамматики

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	2	2	0	0	0
2	2	2	1	0	0
3	0	1	0	0	0
4	3	1	0	0	0
5	2	1	1	0	0
6	0	0	0	0	0
7	0	1	1	0	0
8	0	0	0	0	0
9	0	0	1	0	0
10	1	0	1	0	0
11	1	0	0	0	0
12	0	0	0	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	0	0
16	1	1	1	0	0
17	0	0	0	0	0
18	1	1	1	0	0
19	1	0	1	1	0
20	0	0	0	0	0
21	0	0	0	0	0
22	0	0	0	0	0
23	0	0	0	0	0
24	1	0	1	0	0
25	1	1	1	1	0
26	0	1	0	0	0
27	0	0	0	0	0
28	0	0	0	0	0
29	3	2	0	0	0
30	1	1	1	0	0

Продолжение таблицы 41

31	1	0	0	1	0
32	0	0	0	0	0
33	0	0	1	0	0
34	1	0	1	0	0
35	2	0	1	0	0
36	0	0	0	0	0
37	0	0	0	0	0
38	0	0	0	0	0
39	0	0	0	0	0
40	1	1	1	1	0
41	2	1	0	2	0
42	0	0	0	0	0
43	1	0	1	0	0
44	1	1	1	0	0
45	1	1	1	0	0
46	0	2	1	0	0
47	0	0	0	0	0
48	0	3	1	0	0
49	1	1	0	0	0
50	2	1	1	1	0

Ключевые выражения, извлеченные экстрактором, представляют собой существительные, сочетания существительных с существительными, а также существительных и прилагательных. Для публицистического подкорпуса максимальное среднее совпадений составило 0,66, что является довольно высоким результатом, учитывая факт отсутствия в грамматике правил с глаголами, предложными конструкциями и другими частями речи.

Таблица 42 – Количество совпадений извлеченных ключевых выражений в научном подкорпусе, проанализированном экспертами экстрактором на основе грамматики

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	1	0	0	0	0
2	3	2	0	1	0
3	0	1	1	0	0
4	1	0	0	0	0
5	2	0	0	0	0
6	0	0	0	0	0
7	2	0	0	1	0

Продолжение таблицы 42

8	0	0	1	0	0
9	1	0	0	0	0
10	2	2	0	0	0
11	0	1	0	0	0
12	1	1	1	1	0
13	0	1	0	0	0
14	1	0	1	0	0
15	1	0	0	1	0
16	1	2	1	0	0
17	2	0	1	1	1
18	0	0	1	0	0
19	0	0	0	0	0
20	2	3	0	0	0
21	0	0	0	0	0
22	1	0	1	0	0
23	3	0	0	0	0
24	1	0	0	0	0
25	0	0	0	0	0
26	2	1	0	1	0
27	0	0	0	0	0
28	2	0	0	0	1
29	1	1	0	0	0
30	1	1	0	0	0
31	3	0	2	0	0
32	2	1	0	1	0
33	1	1	0	0	0
34	3	0	1	0	0
35	2	1	0	0	0
36	1	0	0	0	0
37	1	0	0	0	0
38	1	1	0	0	0
39	1	1	0	0	0
40	0	0	0	0	0
41	1	0	0	0	0
42	0	0	0	0	0
43	0	0	0	0	0
44	1	0	1	0	0
45	1	0	0	0	0
46	1	1	1	0	0
47	1	2	2	1	0
48	0	1	0	0	0

Продолжение таблицы 42

49	1	0	1	0	0
50	0	0	0	0	0

Для научного подкорпуса экстрактор справился лучше всех, что объясняется моделями, созданными на основе ключевых выражений научного подкорпуса. Максимальное среднее совпадений равно 1,04 слова на текст, что может считаться довольно высоким результатом для данного экстрактора.

Таблица 43 – Количество совпадений извлеченных ключевых выражений в художественном подкорпусе, проанализированном экспертами экстрактором на основе грамматики

№	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
1	2	0	1	0	0
2	0	0	0	0	0
3	2	1	1	0	0
4	1	0	0	0	0
5	1	0	0	0	0
6	1	0	0	0	0
7	2	0	1	0	0
8	0	0	0	0	0
9	2	1	0	0	0
10	0	0	1	0	0
11	1	0	0	0	0
12	1	2	1	0	0
13	0	0	0	0	0
14	1	2	0	0	0
15	0	0	0	0	0
16	0	0	0	0	0
17	1	0	1	0	0
18	1	0	0	1	0
19	0	0	0	0	0
20	2	0	0	0	0
21	0	0	0	0	0
22	1	0	0	0	0
23	0	1	0	0	0
24	0	0	0	0	0
25	0	1	0	0	0
26	0	1	0	0	0
27	0	0	0	0	0

Продолжение таблицы 43

28	1	0	0	0	0
29	0	0	0	0	0
30	0	0	0	0	0
31	0	1	0	0	0
32	0	0	0	0	0
33	1	0	0	0	0
34	1	0	0	0	0
35	0	1	0	0	0
36	0	0	0	0	0
37	0	0	0	0	0
38	0	0	0	0	0
39	0	0	0	0	0
40	1	2	1	1	0
41	0	0	0	0	0
42	1	1	0	0	0
43	1	2	0	0	0
44	0	1	1	0	0
45	0	1	1	0	0
46	0	0	0	0	0
47	0	1	0	0	0
48	1	1	0	0	0
49	0	0	0	0	0
50	0	0	0	0	0

Среди большого множества ключевых выражений художественного подкорпуса экстрактора нашлось не слишком много совпадений с экспертами. Однако ключевые выражения экстрактора все же отражают сюжет и повествование событий в главах текстов при помощи названия объектов, имен действующих лиц, предметов взаимодействия, писания окружения, пейзажа и природы. Максимальное среднее совпадений составило 0,52 слова на текст.

В таблице 44 приведены результаты совпадений работы экстрактора и экспертов: сумма и средний показатель по каждому из подкорпусов. Максимальный показатель среднего совпадений получился у эксперта номер 1 для научного подкорпуса, минимальные показатели у эксперта номер 5 для публицистического и художественного подкорпусов. Значения показателей

равны 0. Остальные результаты находятся в пределах максимального и минимального значений.

На рисунке 5 приведена диаграмма совпадений работы экстрактора и экспертов для всех трех подкорпусов.

Таблица 44 – Результаты оценки совпадений работы экстрактора и экспертного решения в публицистическом, научном и художественном подкорпусах

Алгоритм	Подкорпус	Показатель	ЭКС1	ЭКС2	ЭКС3	ЭКС4	ЭКС5
Экстрактор	Публицистический	сумма	33	26	21	7	0
		среднее	0,66	0,52	0,42	0,14	0
	Научный	сумма	52	25	16	8	2
		среднее	1,04	0,5	0,32	0,16	0,04
	Художественный	сумма	26	20	9	2	0
		среднее	0,52	0,4	0,18	0,04	0

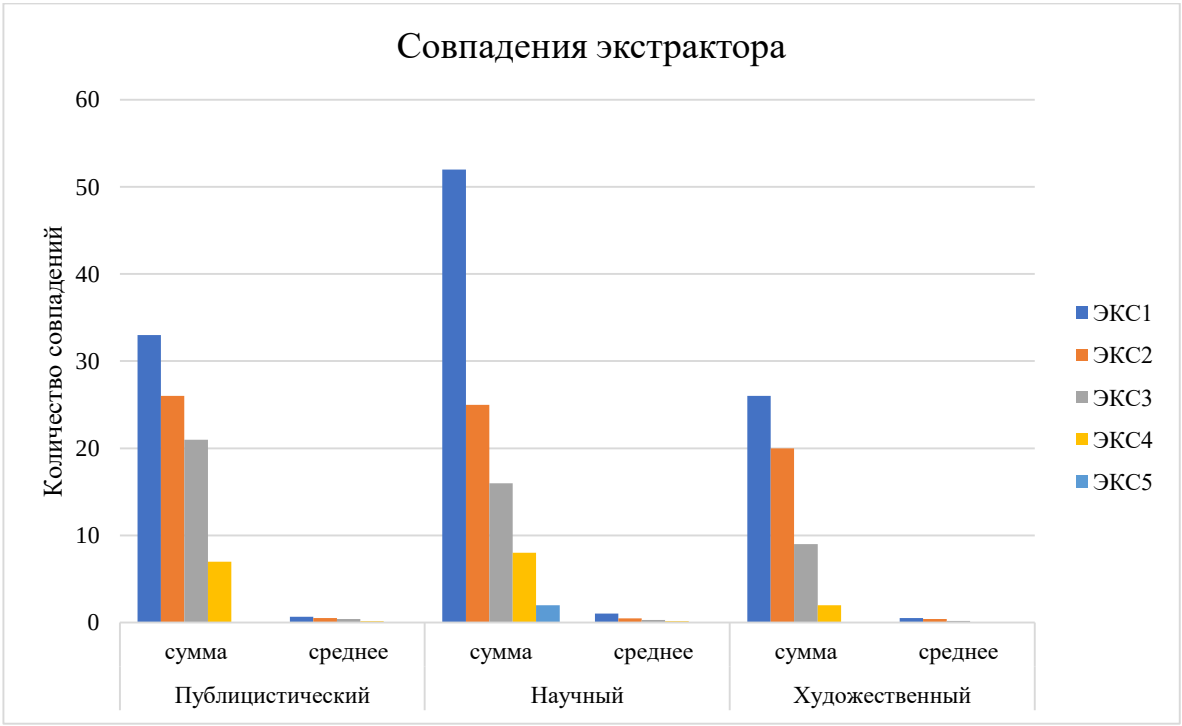

Рисунок 5. Диаграмма совпадений экстрактора для публицистического, научного и художественного подкорпусов

7.4. Проведение расчетов оценки результатов экстрактора

Оценка результатов работы экстрактора проводилась с использованием формул из раздела 6 настоящей работы. Таким образом, точность, полнота и F-мера экстрактора для всех трех подкорпусов равны.

Точность для экстрактора публицистического подкорпуса (91):

$$P = \frac{87}{87 + 522} = 0{,}142857143 \quad (91)$$

Полнота для экстрактора публицистического подкорпуса (92):

$$R = \frac{87}{87 + 697} = 0{,}110969388 \quad (92)$$

F-мера для экстрактора публицистического подкорпуса (93):

$$F = 2 * \frac{0{,}142857143 * 0{,}110969388}{0{,}142857143 + 0{,}110969388} = 0{,}124910266 \quad (93)$$

Точность для экстрактора научного подкорпуса (94):

$$P = \frac{103}{103 + 1978} = 0{,}049495435 \quad (94)$$

Полнота для экстрактора научного подкорпуса (95):

$$R = \frac{103}{103 + 680} = 0{,}131545338 \quad (95)$$

F-мера для экстрактора научного подкорпуса (96):

$$F = 2 * \frac{0{,}049495435 * 0{,}131545338}{0{,}049495435 + 0{,}131545338} = 0{,}071927374 \quad (96)$$

Точность для экстрактора художественного подкорпуса (97):

$$P = \frac{57}{57 + 4276} = 0{,}013154858 \quad (97)$$

Полнота для экстрактора художественного подкорпуса (98):

$$R = \frac{57}{57 + 666} = 0{,}078838174 \quad (98)$$

F-мера для экстрактора художественного подкорпуса (99):

$$F = 2 * \frac{0{,}013154858 * 0{,}078838174}{0{,}013154858 + 0{,}078838174} = 0{,}022547468 \quad (99)$$

Оценка эффективности экстрактора ключевых выражений может быть признана средней относительно результатов оценки других алгоритмов. Для экстрактора показатели резко уменьшаются только для художественного подкорпуса. Таким образом, f-мера для публицистического подкоропуса составила 12,5%, для научного – 7%, для художественного – 2%.

Итак, результатам оценки эффективности можно заключить, что разработанный нами экстрактор, основанный на грамматике, имеет схожие, а иногда даже и превосходящие показатели оценки ключевых выражений, чем традиционные алгоритмы.

Заключение

В ходе исследования была достигнута следующая цель: экспериментальным путем было определено, что существует зависимость местоположения ключевых выражений относительно всего текста при помощи сравнения экспертной разметки и различных методов автоматического выделения ключевых выражений при работе с русскоязычными текстами различной тематики и стилей. Ключевые выражения содержатся в самом начале текста и с малой вероятностью появляются в основной части и заключении документа.

В настоящей работе было проведено исследование природы ключевых выражений относительно структуры текста. Гипотеза о существовании зависимости местоположения ключевого выражения подтвердилась в ходе эксперимента, который состоял в том, чтобы поделить текст на две части, извлечь ключевые выражения из начала при помощи экспертов, а при помощи алгоритмов из оставшейся части и сравнить количество совпадений. Суммы совпадений оказались настолько низкими, что был сделан вывод о концентрации ключевых выражений в самом начале текста. Оценка эффективности алгоритмов не поднялась выше 0,2, что говорит о малом количестве встречаемости ключевых выражений результатов алгоритмов и разметки экспертов.

Важно отметить тот факт, что для подкорпусов всех трех функциональных стилей – публицистического, научного и художественного – наблюдается подтверждение гипотезы. Самые высокие показатели совпадений получились у публицистического подкорпуса, самые низкие – у художественного. Такая ситуация объясняется простой и краткостью новостных текстов публицистического подкорпуса и объемом и сложностью документов художественного подкорпуса.

В ходе работы над диссертацией был создан и опробован собственный экстрактор ключевых выражений, основанный на грамматике. Данный алгоритм

способен извлекать выражения, нужные пользователю, то есть необходимо написать правила грамматики, и алгоритм начнет работу.

Для настоящего исследования были написаны правила грамматики, основанные на ключевых выражениях научного подкорпуса. Экстрактор наравне с другими традиционными алгоритмами извлекал ключевые выражения. Результаты оценки эффективности были признаны средними относительно остальных алгоритмов, однако, опровергнуть гипотезу разработанному экстрактору не удалось, показатели совпадений и f-меры по-прежнему остались в пределах от 0 до 0,2.

Список источников

1. Браславский, П.И. Автоматическое извлечение терминологии с использованием поисковых машин интернета / Е.А. Соколов; [Электронный ресурс / Electronic resource] : научная статья. – Режим доступа : http://www.dialog-21.ru/digests/dialog2007/materials/html/14.htm
2. Ванюшкин, А.С. Методы и алгоритмы извлечения ключевых слов / Л.А. Гращенко; Новые информационные технологии в автоматизированных системах. 2016. №19. С. 85-87.
3. Гамзатова, А.Ф. «Эмоциональное» и «формальное»: проблема выделения ключевых слов компьютерными программами в сопоставлении с методикой их экспертного вычленения.
4. Захаров, В.П. Анализ эффективности статистических методов выявления коллокаций в текстах на русском языке. – 2010. – / М.В. Хохлова; Компьютерная лингвистика и интеллектуальные технологии: Труды международной конференции «Диалог – 2010».
5. Москвина, А. Д. Автоматическое выделение ключевых слов и словосочетаний из русскоязычных корпусов текстов с помощью алгоритма RAKE / О. А. Митрофанова, А. Р. Ерофеева, Я. К. Харабет; Труды международной конференции «Корпусная лингвистика – 2017». –СПб: Издательство Санкт-Петербургского университета, 2017. – С. 268–275.
6. Москвитина, Т.Н. Ключевые слова и их функции в научном тексте // Вестник Челябинского государственного педагогического университета. 2009. № 11. С. 270-283.
7. Мурзин, Л. Н. Текст и его восприятие / А. С. Штерн; Свердловск : Изд-во Урал. ун-та, 1991.
8. Система PullEnti - извлечение информации из текстов естественного языка и автоматизированное построение информационных систем / О. В. Золотарев, М. М. Шарнин, С. В. Клименко, К. И. Кузнецов // Ситуационные центры и информационно-аналитические системы класса 4i для задач

мониторинга и безопасности (SCVRT2015-16) : Труды Международной научной конференции: в 2-х томах, ЦарьГрад, Московская область, Россия, 21–24 ноября 2016 года. – ЦарьГрад, Московская область, Россия: Автономная некоммерческая организация "Институт физико-технической информатики", 2016. – С. 28-35.

9. Усталов, Д.А. Извлечение терминов из русскоязычных текстов при помощи графовых моделей. – 2012. – // CSEDays: Теория графов и приложения. – Екатеринбург.
10. Шереметьева, С.О. Методы и модели автоматического извлечения ключевых слов / С.О. Шереметьева, П.Г. Осминин // Вестник ЮУрГУ. Серия «Лингвистика» : 2015. – Т. 12, № 1. – С. 76–81.
11. Ягунова, Е.В. Эксперимент и вычисления в анализе ключевых слов художественного текста // Философия языка. Лингвистика. Лингводидактика №1 Пермь : 2010. с.83-89
12. Campos, R., Mangaravite, V., Pasquali, A., Jorge, A.M., Nunes, C., Jatowt, A.: Yake! collection-independent automatic keyword extractor. In: European Conference on Information Retrieval. Springer : 2018. pp. 806–810.
13. Campos, R., Mangaravite, V., Pasquali, A., Jatowt, A., Jorge, A., Nunes, C. and Jatowt, A. (2020). YAKE! Keyword Extraction from Single Documents using Multiple Local Features. In Information Sciences Journal. Elsevier, Vol 509, pp 257-289.
14. Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Domain-specific keyphrase extraction. In Proceedings of 16th International Joint Conference on Artificial Intelligence, pages 668–673.
15. Kazi Saidul Hasan and Vincent Ng, Automatic Keyphrase Extraction: A Survey of the State of the Art. ACL : 2014.
16. KeyBERT [Электронный ресурс] : статья. – Режим доступа : https://blog.google/products/search/search-language-understanding-bert/.
17. Luhn H.P. A Statistical Approach to Mechanized Encoding and Searching of Literary Information // IBM J. Res. Dev. №4. - 1957 . - С. 309–317.

18. Luhn H.P. The Automatic Creation of Literature Abstracts // IBM J. Res. Dev. - 1958. - April- C. 159–165.
19. Maarten Grootendorst: KeyBERT: Minimal keyword extraction with BERT. // - 2020. Zenodo: [Электронный ресурс] : статья. – Режим доступа : https://doi.org/10.5281/zenodo.4461265.
20. Matzuo Y., Ishizuka M. Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information // Int. J. Artificial Intell. Tools. - 2004 . - C. 13.
21. Olena Medelyan, Eibe Frank, and Ian H. Witten. 2009. Human-competitive tagging using automatic keyphrase extraction. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 1318–1327.
22. Peter Turney. 2003. Coherent keyphrase extraction via web mining. In Proceedings of the 18th International Joint Conference on Artificial Intelligence, pages 434–439.
23. Pymorphy2 [Электронный ресурс] : открытое программное обеспечение. – Режим доступа : https://pymorphy2.readthedocs.io/en/stable/.
24. Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into texts. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pages 404–411.
25. Rose S.J., Cowley W.E., Crow V., и др. Rapid Automatic Keyword Extraction for Information Retrieval and Analysis // . - 2011. - Т. 1. - № 19.
26. rutermextract [Электронный ресурс] : открытое программное обеспечение. – Режим доступа : https://github.com/igor-shevchenko/rutermextract.
27. SDK PullEnti [Электронный ресурс] : открытое программное обеспечение. – Режим доступа : https://www.pullenti.ru/.
28. TextRank — NLPub [Электронный ресурс] : приложение к алгоритму PageRank : http://nlpub.ru/TextRank.

29. Tomokiyo T., Hurst M. A language model approach to keyphrase extraction // Proc. ACL 2003 Work. Multiword expressions Anal. Acquis. Treat. -. - 2003. - T. 18 . - C. 33–40.
30. topia.termextract 1.1.0 [Электронный ресурс] : открытое программное обеспечение. – Режим доступа : https://pypi.org/project/topia.termextract/.
31. V´ıt Baisa, V´ıt Suchomel. Large Corpora for Turkic Languages and Unsupervised Morphological Analysis.

Приложение А. Листинг программы для автоматической сборки корпуса

```
import requests
from bs4 import BeautifulSoup

r = requests.get ('https:// meduza.io/')
print (1)
if r.status_code != 200:
    print('Wrong!')

soup = BeautifulSoup (r.content, features='html.parser')
news = soup.findAll ('div', {'class':'capt'})

base_link = 'https://littleone.com'

news_file = open ('KW.txt', 'w', encoding = 'utf-8')

for a1 in news:
    link = a1.find('a', href = True) ['href']
    print(link)
    curr_news = requests.post(base_link + link)
    first_news_soup = BeautifulSoup(curr_news.content, features='html.parser')
    article = first_news_soup.findAll('p', {'class':'public_block'})
    for paragr in article:
        news_file.write(paragr.text)
        news_file.write('\n')
    news_file.write('\n')

news_file.close()
```

Приложение Б. Листинг программы для сборки ключевых выражений из научных статей

```
import re

f = open('kw_naked.txt', 'r', encoding = 'UTF-8')
f1 = open('kw_strings.txt', 'w', encoding = 'UTF-8')
for s in f:
    #s = "русский язык, дискурсивные единицы, паузы, корпус устной речи"
    result = re.split(r'(?<!\d),|,(?!\d)', s)
    for i in range(len(result)):
        print(result[i], file=f1)

f1.close()
###
f = open('kw_strings.txt', 'r', encoding = 'UTF-8')
f1 = open('kw_strings1.txt', 'w', encoding = 'UTF-8')
f2 = open('kw_strings2.txt', 'w', encoding = 'UTF-8')
f3 = open('kw_strings3.txt', 'w', encoding = 'UTF-8')
f4 = open('kw_strings4.txt', 'w', encoding = 'UTF-8')
for s in f:
    #s = input()
    s = s.split()
    l = len(s)
    if l == 1:
        print(s, file=f1)
    if l == 2:
        print(s, file=f2)
    if l == 3:
        print(s, file=f3)
```

```
    if l >= 4:
        print(s, file=f4)

f1.close()
f2.close()
f3.close()
f4.close()
###
f = open('kw_3_results.txt', 'r', encoding = 'UTF-8')

i = 0
k=0
l = 0
m=0
n=0
p=0
j =0
b= 0

for strings in f:
    if "['ADJF,nomn,sing', 'ADJF,nomn,sing', 'NOUN,nomn,sing']" in strings:
        i = i+1
    if "['ADJF,nomn,plur', 'NOUN,nomn,plur', 'NOUN,gent,plur']"  in strings:
        k = k+1
    if "['NOUN,nomn,sing', 'ADJF,gent,sing', 'NOUN,gent,sing']"  in strings:
        l = l+1
    if "['NOUN,nomn,plur', 'ADJF,gent,sing', 'NOUN,gent,sing']"  in strings:
        m = m+1
    if "['NOUN,nomn,sing', 'ADJF,gent,plur', 'NOUN,gent,plur']"  in strings:
        n = n+1
```

```
    if "['ADJF,nomn,sing', 'NOUN,nomn,sing', 'NOUN,gent,plur']"  in strings:
        p = p+1
    if "['ADJF,nomn,sing', 'NOUN,nomn,sing', 'NOUN,gent,sing']"  in strings:
        j = j+1
    if "['ADJF,nomn,plur', 'NOUN,nomn,plur', 'NOUN,gent,plur']"  in strings:
        b = b+1


print('kw sing:', i)
print('kw plur:', k)
print('kw plur:', l)
print('kw sing:', m)
print('kw plur:', n)
print('kw plur:', p)
print('kw plur:', j)
print('kw plur:', b)
```

Приложение В. Листинг программы для морфологического анализа слов и словосочетаний

```
# Описания шаблонов
source = '''
сущ
NOUN,nomn,sing
сущ
NOUN,nomn,plur
прил
ADJF,nomn,sing
сущ прил
ADJF,nomn,sing NOUN,nomn,sing
сущ прил
ADJF,nomn,plur NOUN,nomn,plur
сущ сущ
NOUN,nomn,sing NOUN,gent,plur
сущ сущ
NOUN,nomn,plur NOUN,gent,plur
сущ сущ
NOUN,nomn,sing NOUN,gent,sing
сущ сущ
NOUN,nomn,plur NOUN,gent,sing
прил прил сущ
ADJF,nomn,sing ADJF,nomn,sing NOUN,nomn,sing
прил сущ сущ
ADJF,nomn,plur NOUN,nomn,plur NOUN,gent,plur
сущ прил сущ
NOUN,nomn,sing ADJF,gent,sing  NOUN,gent,sing
сущ прил сущ
```

NOUN,nomn,plur ADJF,gent,sing NOUN,gent,sing

сущ прил сущ

NOUN,nomn,sing ADJF,gent,plur NOUN,gent,plur

прил сущ сущ

ADJF,nomn,sing NOUN,nomn,sing NOUN,gent,plur

прил сущ сущ

ADJF,nomn,sing NOUN,nomn,sing NOUN,gent,sing

прил сущ сущ

ADJF,nomn,plur NOUN,nomn,plur NOUN,gent,sing

прил прил сущ

ADJF,nomn,plur ADJF,nomn,plur NOUN,nomn,plur

'''

text = '''

62 КС

Русский ассоциативный словарь

Корпус повседневной речи

анализ формальных понятий

Русский жестовый язык

корпус устной речи

понятие языкового материала

сопоставимые корпуса текстов

параллельные корпуса текстов

сценарии гуманитарного образования

надкорпусные базы данных

Надкорпусные базы данных

Корпус ученических текстов

автоматическая морфологическая разметка

невербальное коммуникативное поведение

матрица совместной встречаемости

русскоязычные корпусы текстов

хранение корпусных данных

поиск в корпусе

разрешение морфологической неоднозначности

русскоязычные корпусы текстов

разметка по ошибкам

Детский билингвизм/детское двуязычие

современный русский язык

повседневная речевая коммуникация

автоматическое исправление ошибок

освоение специальной лексики

теория риторических структур

повседневная разговорная речь

частотность лексических единиц

модели частотного поведения

русскоязычные корпусы текстов

метрики семантической близости

лексическая база данных

компьютерные средства обучения

автоматический анализ текстов

учебник русского языка

извлечение именованных сущностей

семантические классы глагола

тундровый ненецкий язык

FieldWorks Language Explorer

теория риторических структур

Теория риторических структур

предложения с 对

корпус русских документов

микросинтаксическая разметка корпуса
язык международных договоров
прикладной семантический словарь
ручная разметка текста
магическое число Миллера
устойчивые сочетания лексем
средневековые славянские рукописи
Томский диалектный корпус
русские говоры Сибири
ижорские народные песни
автоматическое распознавание речи
язык программирования Python
Сбалансированная аннотированная текстотека
корпус устной речи
повседневная устная речь
биографическая база данных
метрообразующий период стиха
регулятивы стихового метра
'''

```
import pymorphy2 as py

names = {}
morph = py.MorphAnalyzer()

class PPattern:
    def __init__(self):
        super().__init__()
        self.tags = []
        self.rules = []
        self.example = ''
```

```
def checkPhrase(self, words, used = set()):
    def getNextWord(wordList):
        if len(wordList) == 0:
            return None
        index = wordList[0]
        wordList[0:1] = []
        return index

    def checkWord(tags, word, prevResult):
        variants = morph.parse(word)
        for v in variants:
            if set(tags) <= v.tag.grammemes \
               and self.checkRules(prevResult + [(word, v)]):
                return (word, v)
        return None

    allResults = []
    result = []
    wordList = list(set([ x for x in range(0, len(words)) ]) - used)
    wordList.sort()
    wi = getNextWord(wordList)
    nextTag = 0
    usedP = set()
    while wi is not None:
        w = words[wi]
        res = checkWord(self.tags[nextTag].split(','), w, result)
        if res is not None:
            result.append(res)
            usedP.add(wi)
```

```
                nextTag = nextTag + 1
                if nextTag >= len(self.tags):
                    return (result, usedP)
            wi = getNextWord(wordList)
        return None

    def checkRules(self, result):
        for r in self.rules:
            indexes = r[0]
            func = r[1]
            l = r[2]
            if max(indexes) < len(result): # У нас есть достаточно данных
                args = [ result[x][1] for x in indexes ]
                if not func(*args):
                    return False
        return True

    def checkPropRule(self, getFunc, getArgs, srcFunc, srcArgs, \
                      op = lambda x,y: x == y):
        v1 = getFunc(getArgs)
        v2 = srcFunc(srcArgs)
        return op(v1,v2)

    def setProp(self, setFunc, setArgs, srcFunc, srcArgs):
        setFunc(setArgs, srcFunc(srcArgs))

import io
import ast

def parseSource(src):
```

```
def parseFunc(expr, names):
    m = ast.parse(expr)
    # Получим список уникальных задействованных имен
    varList = list(set([ x.id for x in ast.walk(m) if type(x) == ast.Name]))
    # Найдем их позиции в грамматике
    indexes = [ names.index(v) for v in varList ]
    lam = 'lambda %s: %s' % (','.join(varList), expr)
    return (indexes, eval(lam), lam)
def parseLine(s):
    nonlocal arr, last
    s = s.strip()
    if s == '':
        last = None
        return
    if s[0] == '#':
        return
    if last is None:
        last = PPattern()
        arr.append(last)
    if s[0] == ':': # имена
        names[s[1:]] = last
    elif s[0] == '-': # внутренние имена
        s = [x.strip('-') for x in s[1:].strip().split()]
        last.names = s
    elif s[0] == '=': # правила
        expr = s[1:].strip()
        last.rules.append(parseFunc(expr, last.names))
    else:
        last.tags = s.split()
```

```
    arr = []
    last = None
    buf = io.StringIO(src)
    s = buf.readline()
    while s:
        parseLine(s)
        s = buf.readline()
    return arr

def parseText(pats, text):
    def parseLine(line):
        words = line.split()
        allSet = set([x for x in range(len(words))])
        used = set()
        was = False
        for p in pats:
            usedP = set()
            while True:
                res = p.checkPhrase(words, usedP)
                if res:
                    (res, newP) = res
                    used = used.union(newP)
                    first = list(newP)[0]
                    usedP = set([ x for x in range(first+1)])
                    print('+',line, p.tags, [r[0] for r in res])
                    was = True
                else:
                    break
        if not was:
            print('-',line)
```

```
    buf = io.StringIO(text)
    s = buf.readline()
    while s:
        s = s.strip()
        if s != '':
            parseLine(s)
        s = buf.readline()

def tags(word):
    morph = py.MorphAnalyzer()
    return morph.parse(word)

patterns = parseSource(source)
parseText(patterns, text)
```

Приложение Г. Листинг программы для экстрактора ключевых выражений на основе грамматики

```
from pymorphy2 import MorphAnalyzer
from hmmtrigram import MostProbableTagSequence
from nltk import word_tokenize


class POStagger:
    def __init__(self, file_name):
        self.morph = MorphAnalyzer()
        self.mps = MostProbableTagSequence(file_name)
        self.end_of_sentence_markers = ['.', '!', '?', '\n']

    def parse(self, file_name):
        with open(file_name, 'r', encoding='UTF-8') as reader:
            txt_file = reader.read()
        tokenized_text = word_tokenize(txt_file)
        sentences = self.__sentence_divider(tokenized_text)
        parsed_sentences = self.__morphoanalyzer(sentences)
        most_probable_tags_sequence =
self.__get_most_probable_pos_tag_sequence(parsed_sentences)
        return most_probable_tags_sequence, len(tokenized_text)

    def __sentence_divider(self, tokenized_text):
        sentences = []
        current_sentence = []
        for token in tokenized_text:
            current_sentence.append(token)
            if token in self.end_of_sentence_markers:
```

```
                    sentences.append(current_sentence)
                    current_sentence = []
            return sentences

    def __morphoanalyzer(self, sentences):
            parsed_sentences = []
            for sentence in sentences:
                    parsed_sentence = []
                    for token in sentence:
                            parsed = self.morph.parse(token)
                            parsed_sentence.append(parsed)
                    parsed_sentences.append(parsed_sentence)
            return parsed_sentences

    def __get_most_probable_pos_tag_sequence(self, parsed_sentences):
            most_probable_tags_sequences = []
            for sentence in parsed_sentences:
                    most_probable_tags_seq = self.mps.get_sequence(sentence)
                    most_probable_tags_sequences.append(most_probable_tags_seq)
            return most_probable_tags_sequences


class Keywords(object):
    def __init__(self, file_name):
            self.rules_dict = {}
            with open(file_name) as file:
                    for row in file:
                            if len(row) > 3 and '#' not in row:
                                    rule = self.__validate_form(row)
                                    rule.append('END')
```

```
                        key = ''
                        for i,j in enumerate(rule):
                            if len(rule)-1 > i:
                                key = '{} {}'.format(key, j).strip()
                                if key not in self.rules_dict.keys():
                                    self.rules_dict[key] = []
                                r = rule[i+1].strip()
                                if r not in self.rules_dict[key]:
                                    self.rules_dict[key].append(r)


    def __validate_form(self, row):
        rule = row.strip().split(' ')[::-1]
        composed = ['NOUN', 'ADJF', 'PRTF', 'NPRO', 'NUMR']
        simple = ['ADJS', 'COMP', 'VERB', 'INFN', 'PRTS', 'GRND', 'ADVB',
'PRED', 'PREP', 'CONJ', 'PRCL', 'INTJ']
        number = ['sing', 'plur']
        case = ['nomn', 'gent', 'datv', 'accs', 'ablt', 'loct', 'voct', 'gen1', 'gen2', 'acc2',
'loc1', 'loc2']
        for term in rule:
            if '_' in term:
                tag = term.split('_')
                if (len(tag) != 3) or (tag[0] not in composed or tag[1] not in
number or tag[2] not in case):
                    raise Exception('Invalid tag in rules. Please check:
',term)
            else:
                if term not in simple:
                    raise Exception('Invalid tag in rules. Please check:
',term)
        return rule
```

```
    def __format_key(self, token):
        composed_pos = ['NOUN', 'ADJF', 'PRTF', 'NPRO', 'NUMR']
        pos = token.tag._POS
        if pos in composed_pos:
            key = '{}_{}_{}'.format(pos, token.tag.number, token.tag.case)
        else:
            key = pos
        return key


    def extract(self, tagged_text, token_count, threshold=0.001,
weight_for_chunks=1):
        lemmas_count = {}
        keywords_in_text = []
        keylemmas_in_text = []
        print('\n')
        for sentence in tagged_text:
            keywords_in_sentence = []
            keylemmas_in_sentence = []
            reversed_sentence = sentence[::-1]
            sentence_length = len(reversed_sentence)
            i = 0
            while i < sentence_length:
                key = self.__format_key(reversed_sentence[i])
                word = reversed_sentence[i].word
                lemma = reversed_sentence[i].normal_form
                if lemma not in lemmas_count.keys():
                    lemmas_count[lemma] = 0
                lemmas_count[lemma] += 1
                if key in self.rules_dict.keys():
                    c = 1
```

```
                while key in self.rules_dict.keys():
                    value = self.rules_dict[key]
                    if 'END' in value:
                        keywords_in_sentence.append(word)
                        keylemmas_in_sentence.append(lemma)
                    if i+c < sentence_length:
                        next = self.__format_key(reversed_sentence[i+c])
                        if next in value:
                            word = '{} {}'.format(reversed_sentence[i+c].word, word)
                            lemma = '{} {}'.format(reversed_sentence[i+c].normal_form, lemma)
                            key = '{} {}'.format(key, next)
                        else:
                            break
                    else:
                        break
                    c += 1
            i += 1
        if len(keywords_in_sentence) > 0:
            keywords_in_text.append(keywords_in_sentence)
            keylemmas_in_text.append(keylemmas_in_sentence)

    results = self.__set_weights(keywords_in_text, keylemmas_in_text, lemmas_count, token_count, threshold, weight_for_chunks)
    return results

def __set_weights(self, keywords_in_text, keylemmas_in_text, lemmas_count, token_count, threshold, weight_for_chunks):
    weighted_chunks = {}
```

```
        for i, sentence in enumerate(keywords_in_text):
            for j, chunk in enumerate(sentence):
                weighted_chunks[keywords_in_text[i][j]] = 0
                chunk_elements = chunk.split(' ')
                for element in chunk_elements:
                    if element in lemmas_count.keys():
                        weighted_chunks[keywords_in_text[i][j]] += lemmas_count[element]
        for k,v in weighted_chunks.items():
            n_keywords_in_chunk = len(k.split(' '))
            #weighted_chunks[k] = (v / token_count) * (n_keywords_in_chunk ** weight_for_chunks)
            weighted_chunks[k] = (v * (n_keywords_in_chunk ** weight_for_chunks)) / token_count

        new_dict = {}
        for k,v in weighted_chunks.items():
            if v > threshold:
                new_dict[k] = v
        sorted_dict = {r: new_dict[r] for r in sorted(new_dict, key=new_dict.get, reverse=True)}

        return sorted_dict
```