# Набор данных библиотеки научных предметных областей SciLibRu

**О.М.Атаева[1], Н.П.Тучкова[1], К.Б.Теймуразов[1], А.Абдышев[2], М.Г.Кобук[2]**

*[1]ФИЦ ИУ РАН, ул. Вавилова, 40, 119333 Москва, Россия*
*[2]ВУИМ, 2-й Кожуховский проезд, 12с1, 115432 Москва, Россия*

**Аннотация.** В работе дается описание данных предметных областей, составляющих контент семантической библиотеки. Исследование направлено на решение проблемы создания датасета для обучения больших языковых моделей для работы с русскоязычными научными предметными областями. Показаны процедуры наполнения библиотеки публикациями после обработки слабоструктурированных данных, проверки качества графа знаний и навигации по данным. Приложение работы заключается в создании рекомендательных систем для работы с научными русскоязычными журналами.

**Ключевые слова:** прикладная онтология, граф знаний, источники данных, разработка онтологии, научные результаты в цифровой среде

# Dataset of the library of scientific subject areas SciLibRu

**O.M.Ataeva[1]**[0000-0003-0367-5575]**, N.P.Tuchkova[2]**[0000-0001-5357-9640]
**K.B.Teymurazov[3], A.Abdyshev[4], M.G.Kobuk[5]**

*[1,2,3]FRC CSC RAS, Vavilov str., 40, Moscow, 119333, Russia*
*[4,5]VUIM 2nd Kozhukhovsky proezd 12(1), Moscow, 115432, Russia*
[1]OAtaeva@frccsc.ru, [2]NTuchkova@frccsc.ru, [3]KTeymurazov@frccsc.ru,
[4]abdyshovaidin@gmail.com, [5]mikhail.kobuk@mail.ru

**Abstract.** The paper describes the subject area data that make up the content of the semantic library. The study is aimed at solving the problem of creating a dataset for training large language models to work with Russian-language scientific subject areas. The procedures for filling the library with publications after processing weakly structured data, checking the quality of the knowledge graph and navigating through the data are shown. The application of the work is to create recommender systems for working with scientific Russian-language journals.

**Keywords:** applied ontology, knowledge graph, data sources, ontology development, scientific results in the digital environment

## 1. Введение

Описание наборов данных составляет необходимую часть организации доступа к информационным ресурсам. Библиотека предметных областей *SciLibRu* сформирована в рамках контента семантической библиотеки *LibMeta* [1], на основе онтологического проектирования [2] и навигации по данным с помощью графа знаний (knowledge graph, *KG*) [3].

*Семантическая цифровая библиотека LibMeta* – это цифровая библиотека, данные которой связаны иерархическими и ассоциативными отношениями в соответствии с *онтологией предметной области*.

*Онтология цифровой семантической библиотеки* – формальное описание множества данных (типы данных, связи) предметной области (SjD) [4]. Для определения онтологии используется язык OWL(https://www.w3.org/2006/04/OWL_UseCases-ru.html). Онтология на OWL – это RDF-граф (https://www.w3.org/TR/rdf12-schema/). Описание на OWL – это описание *структуры* данных, а *не самих данных*. Каждый экземпляр данных – это экземпляр элемента онтологии. В библиотеке LibMeta *тезаурус* [5] представлен *онтологией* на OWL.

Средства редактирования *LibMeta* – это *средства редактирования онтологии*.

В результате интеграции данных в библиотеке *LibMeta* накоплены описания *SjD* математики и смежных областей, включая описания классификаторов, научных журналов, и других источников, которые семантически связаны в *онтологии библиотеки SjD SciLibRu*. Развитие навигации в библиотеке с применением *KG*, позволяет перейти к поиску, задавая в поисковом запросе принадлежность данных *SjD*.

В нашей работе предлагается описание коллекции данных *SciLibRu*, которые отличаются от данных *LibMeta* наличием новых связей, полученных в результате *метрического анализа* [6, 7] *KG LibMeta* и достраивания онтологии, хотя эти множества, конечно, пересекаются. Описание данных *SciLibRu* стало необходимым исследованием, поскольку дальнейшее развитие библиотеки предполагает постановку задач о применении больших языковых моделей (*LLM*) и создании рекомендаций для извлечения знаний, накопленных в библиотеке. Множество данных, предварительно обученной на общих данных *LLM,* предполагается дополнять данными *KG SciLibRu*, чтобы ограничивать *LLM* рамками предметной области, задаваемой *KG*.

В работе представлено описание семантической модели данных и *KG,* оценка качества данных для интегрированных в библиотеку энциклопедий, пример интеграции *LLM* и *KG* для *SjD* обыкновенных дифференциальных уравнений (*ODE*), описана процедура гибридного алгоритма загрузки слабоструктурированных данных журналов, содержащих символьную информацию (формулы).

## 2. Семантическая модель и данные SciLibRu

### *2.1. Семантическая модель SciLibRu*

Будем использовать понятия *тезауруса* предметной области [8, 9], *онтологии* [4, 7], *KG* (*графа знаний*) [10, 11], *знаний* [12, 13]. *Данные SciLibRu* – это научные статьи, энциклопедии, классификаторы, словари, тезаурусы, корпуса текстов и другие источники оцифрованной информации, представленные в виде онтологии *SjD*. *Онтология SciLibRu* – это семантическая модель данных, в основе которой лежит тезаурус предметной области. Онтология проецируется на *KG*, поскольку реализуется через схему RDF. Основной контент *SciLibRu* составляют математические *SjD*, приложения и смежные области, задаваемые данными прикладных и междисциплинарных журналов.

*Знания* - это структурированные *данные*. Извлечение знаний – это получение ответа на информационный запрос. Ответ может быть (не)релевантный теме запроса и (не)пертинентный, то есть (не) удовлетворяющим информационную потребность пользователя, (не) соответствующий тезаурусу адресата [8].

*Семантическая модель данных* - структурированные данные, где есть *объект*, *субъект* и *отношения* между ними. Именно отношения составляют смысл этой тройки данных. В *KG* отношения – это связи графа, объекты и субъекты – вершины графа.

Онтология *LibMeta* [1] – трехуровневая, содержит:

- универсальные *понятия* онтологии;
- описания *объектов* прикладной области;
- *метаданные прикладной области* как таковые.

Основными элементами онтологии OWL являются описания классов, их свойств, отношений между классами и представителями (индивидов) классов (их свойств и отношений). Для этих описаний OWL последовательно использует бинарные отношения своего словаря, а также словарей RDF и RDFS (https://www.w3.org/TR/owl-semantics/).

*SjD LibMeta*, *DLib* = {*Per, Pub, Th, Enc, Jour*}:

- данные персон и их свойств, множество {*Per*};
- данные публикаций и их свойств, множество {*Pub*};
- данные тезаурусов, множество {*Th*};
- данные энциклопедий, множество {*Enc*};
- данные журналов, множество {*Jour*}.

Данные *SjD SciLibRu DSci* содержат множество *DLib* и данные множества {*KG*}, то есть *DSci* ={*Per, Pub, Th, Enc, Jour, KG*}.

На рис. 1 показано, что к трехуровневой модели онтологии *LibMeta* [1] добавился уровень данных *KG*.

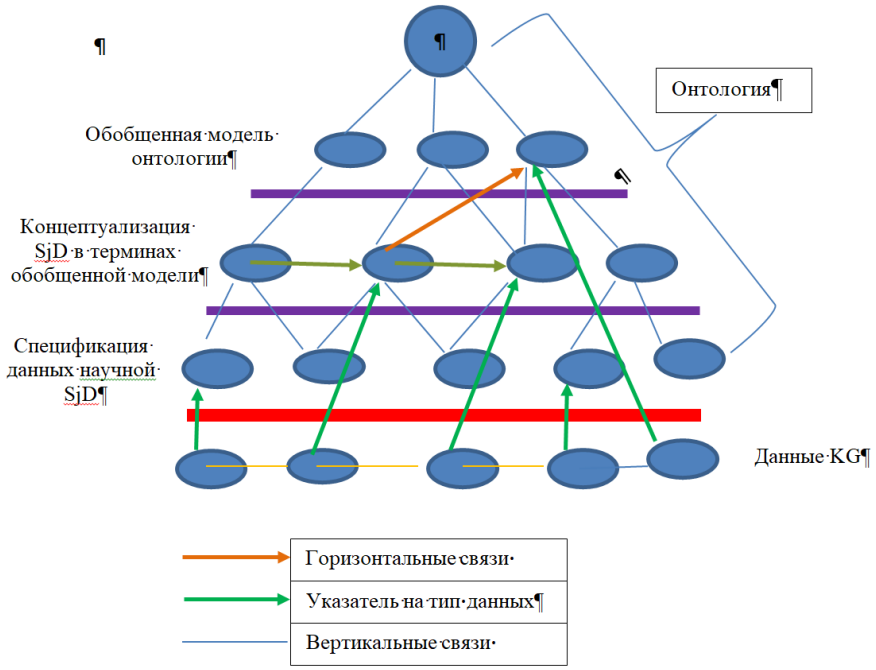


Рис. 1. Архитектура данных в *SciLibRu*.

### *2.2. Качество данных SciLibRu*

Оценка качества данных стала одним из этапов представления знаний, поскольку внедрение *LLM* «перекладывает» вопросы достоверности с *экспертов* научных областей на *создателей* информационных систем, которые организуют интеграцию данных и навигацию по ним. Известные недостатки применения *LLM* в поисковых запросах, которые приводят к «галлюцинациям» связаны как с логикой модели, так и с массивом данных (датасет), на которых проводилось обучение [14].

Для библиотеки *SciLibRu* оценка данных заключается в проверке связей в онтологии, и соответственно, в *KG*, как в результирующем множестве концептов и связей, который будет использоваться на входе *LLM* [15].

Для нашего исследования были посчитаны метрики *KG SciLibRu*, для вершин классификаторов УДК(https://teacode.com/online/udc/) и ГРНТИ(https://grnti.ru/) и получены значения метрик для оценки качества графа.

- Общее *количество узлов* **(**|V|**)** — число вершин (узлов) в графе:

$$|V| = 10813.$$

- Общее *количество связей* **(**|E|**)** — число ребер (связей) в графе:

$$|E| = 178728.$$

- *Распределение узлов KG SciLibRu* по типам, тип 1 - *KG* математической энциклопедии [16], тип 2 – *KG* энциклопедии математической физики [17]:

$$|V|_1 = 6263,$$

$$|V|_2 = 3228.$$

- Распределение узлов *KG SciLibRu* по связям показывает, сколько ассоциативных связей (ребер) каждого типа в KG [1]:

$$|E|_1 = 155448 \text{ (связь } related\text{)},$$

$$|E|_2 = 11805 \text{ (связь } use\text{)},$$

$$|E|_3 = 11475 \text{ (связь } seealso\text{)}.$$

- *Плотность* — мера *заполненности* графа связями:

$$\rho = \frac{|E|}{|V|(|V|-1)}.$$

Для направленного графа: $|V|$ ($|V|$-1) - возможное максимальное количество ребер (каждый узел может иметь ребро к каждому, кроме себя). Плотность графа *KG SciLibRu*: $\rho = 0.0015$.

Низкое значение плотности графа указывает на его *разряженность*, что типично для графов большого размера (граф энциклопедий) и объясняется тем, что большинство узлов связано только с небольшим подмножеством других узлов преимущественно в рамках тематических взаимосвязей каждого понятия.

В графе *KG SciLibRu* выявлено 1057 изолированных узлов типа *Concept*, что составляет около 10% от общего количества понятий *SjD* в графе. Это указывает на отсутствие связей этих понятий с другими сущностями, что может свидетельствовать о недостаточности доступной информации в данной подобласти.

- Также были посчитаны *метрики центральности*

Наиболее *влиятельными* узлами оказались *наиболее общие понятия*. Это свидетельствует о высокой взаимосвязанности и значимости этих понятий в охватываемых *SjD*. Анализ центральности подтвердил значимость этих понятий, при этом выявил дополнительные узлы, которые, несмотря на меньшую распространенность, обладают высокой вероятностной значимостью в графе.

Также в результате анализа были выявлены узлы, замыкающиеся на себя (пример эго-графа «Функция», Рис. 2.), то есть их (концепты и связи) необходимо дополнительно анализировать. В поисковой выдаче, вероятно, это приведет к дублированию.

Качественный анализ графа показал, что есть связи, которые не были выявлены в процессе предобработки данных, что тоже, очевидно, влияет на результат поиска. Была также получена корреляция между эмбеддингами и классическими метриками.

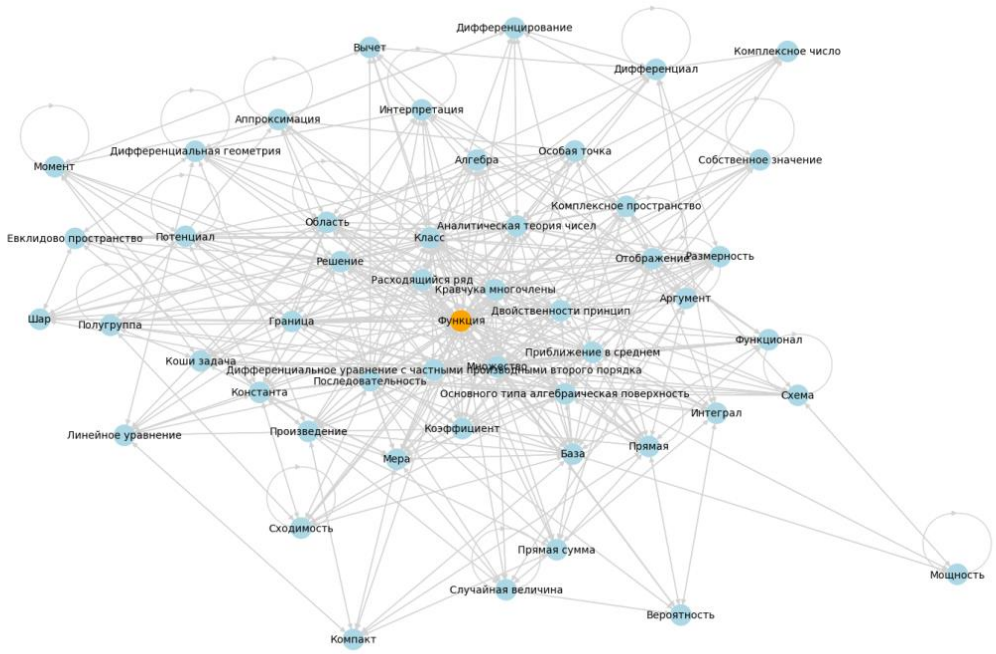


Рис. 2. Эко-граф узла «Функция» (подграф *KG SciLibRu*).

В целом, качественный анализ привел к достраиванию графа и онтологии библиотеки.

### *2.3. Навигация по данным SciLibRu*

Навигация по данным *SciLibRu* реализована с помощью *KG*.

Вершины *KG* (статьи, термины, формулы *SjD*) – экземпляры элементов онтологии, связи – связи тезауруса *SjD*.

Для описания онтологии используются следующие формальные определения:

Объекты и ресурсы:

*OB = {O1, ..., On}, where Oi, i=1,..,n – множество* информационных объектов;

*R = {r1, ..., rm}* , где *rj* , *j=1,…,m – множество* информационных ресурсов (типы информационных объектов);

*TYPE(O)= r - отношение*, где каждому информационному объекту *Oi* соответствует информационный ресурс *rj*;

*IsRe(r1, r2) - отношение*, иерархии ресурсов, где *r2* ($r2 \geq 0$, $r2 \subset r1$) составляет часть ресурса *r1* (подресурс).

Атрибуты:

*A = {a1, ..., ak}- множество* атрибутов информационных ресурсова, *Z*(*ai*) значение атрибута *ai*.

Источники модели данных:

*SX = {(rx, Ar)}, – множество* пар, где *rx* и *Ar* ресурсы и их атрибуты,

*G(rx, Ar) = (r, A), G – отношение*, функция сопоставления элементов

исходной модели данных с информационными ресурсами информационной системы и их набором атрибутов.

Данные источников: *X = (SX, G)*.

Предметная область тезауруса *Th*: *Th = (P, T, R)*, где:

*P - множество* концептов,

*T - множество* вербальных терминов концептов,

*R - множество* вертикальных и горизонтальных связей терминов и концептов.

*F* — функция сопоставления каждого информационного объекта из *OB* с соответствующими терминами из *T*: *F(oi)= {t}, t Є T, oi Є OB.*

Тезаурус *Th* определен для онтологии семантической библиотеки на языке OWL.

*KG* определяется на основе RDF схемы, как триплет (*s, p, o*).

Каждая тройка представляет собой упорядоченный набор терминов RDF:

субъект $s \in U \cup B$,

предикат $p \in U$, и

объект $o \in U \cup B \cup L$.

Кроме того, термин RDF — это URI, $u \in U$, «пустой» узел $b \in B$ или литерал $l \in L$. То есть, *KG* лучше всего представляется на основе схем RDF, но не каждое представление данных RDF рассматривается как *KG*.

Основные связи, которые реализованы между элементами *KG*:

- объект ↔ объект;
- концепт ↔ концепт;
- объект ↔ концепт ↔ объект;
- концепт ↔ объект ↔ концепт;
- классификатор ↔ концепт ↔ классификатор;
- концепт ↔ классификатор ↔ концепт;
- объект ↔ классификатор ↔ объект;
- классификатор ↔ объект ↔ классификатор.

### *2.4. Использование данных SciLibRu*

Множество данных в виде *KG* используется для создания рекомендательных систем с применением *LLM* для коммуникации. Одну из актуальных проблем применения *LLM* можно сформулировать, как *ограничение знаний, на которых формируется ответ LLM при обеспечении полноты данных описания SjD.* Вариантом решения этой задачи может быть использование *KG SjD* в качестве входного множества данных для *LLM* предобученной на общих данных.

В нашем исследовании были проведены эксперименты [18] применения *LLM* к работе с *SjD ODE* в библиотеке *SciLibRu*. Разработана методология взаимодействия *LLM* и *KG SjD ODE* на основе инструкций, применяемых к описанию *SjD* в виде *KG*. На примере *SjD ODE* мы

ограничиваем знания модели *KG SjD*, не позволяя ей выходить за границы *SjD ODE*. На выходе мы получаем ответы на естественном языке, опираясь на знания, представленные в библиотеке в виде *KG*.

В процессе экспериментов были выявлены ошибки, которые влияют на качество данных в библиотеке *SciLibRu*:

1. Синтаксические ошибки в запросах. Это ошибки в генерируемых моделью *LLM* запросах. Например, лишние «слэши» в предикатах, отсутствие префиксов. Для их исправления добавляется инструкция *промпт*.
2. Синтаксические ошибки в онтологии. Они вызывают у модели *LLM* сложности «в понимании», если классы и свойства некорректно именованы или аннотированы. Для их исправления в онтологию добавляется rdfs:label и rdfs:comment и учитывается в дальнейшем при доработке модели данных.

С учетом этих ошибок были сформулированы инструкции к *LLM* с использованием модели *KG SjD ODE*, для формирования SPARQL-запросов по запросам пользователя на естественном языке [18, 19].

## 3. Обновление данных

Обновление данных и дальнейшее наполнение библиотеки, достраивание онтологии и графа знаний реализуется при *интеграции научных статей*, которые проходят предобработку для загрузки в библиотеку. В процессе предобработки данных был разработан *гибридный подход* для интеграции слабоструктурированных данных, содержащих символьную информацию (формулы в LaTeX). Были выявлены проблемы при реализации подхода, которые решались программным путем, написанием соответствующих скриптов на Python. Для удобства изложения выявленные проблемы нумеруются, как П(номер).

Этап предобработки исходных данных журнальных статей для наполнения библиотеки *SciLibRu* потребовал анализа, как самих данных, так и методов конвертации из формата LaTeX, в стандартизированный формат XML, соответствующий заданным спецификациям.

### *3.1. Первичный анализ слабоструктурированных данных*

Архивы выпусков выбранных периодических научных изданий хранятся в формате «ГГГГ-Н.rar» (где ГГГГ — год, Н — номер), обязательно содержат главный исходный файл — IPI.TEX (Рис. 3.).

С помощью файла IPI.TEX выполняется формирование выпуска, при этом редакцией используются существующие шаблоны, обозначения и авторская стилистика применения LaTeX-макрокоманд.

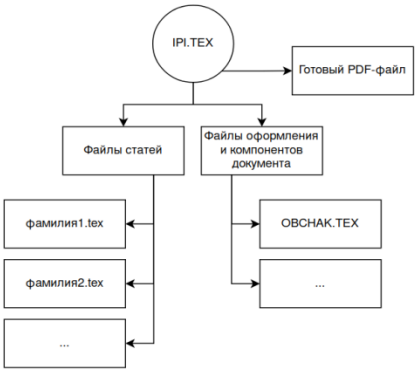


Рис. 3. Структура одного из выпусков журнала выбранного для интеграции

Для выбора стратегии предобработки архивов были проанализированы некоторые подходы и методы, доступные при заданных (ограниченных по времени и мощности) вычислительных ресурсах.

В качестве стратегии для предобработки рассматривались:

***А***. *Синтаксический анализ (парсинг, parsing)* - построение формальной грамматики и использование парсер-генераторов для создания анализатора исходного текста. Этот подход требует тщательной проработки грамматики и может быть сложен в реализации из-за гибкости и расширяемости LaTeX.

***Б***. *Переопределение макрокоманд* - модификация стандартных и пользовательских команд LaTeX для генерации XML-разметки вместо типографского вывода. Этот метод позволяет сохранить семантику документа, но требует глубокого понимания системы макросов LaTeX.

***В***. *Использование промежуточных форматов* - конвертация через промежуточные форматы (DVI, PDF) с последующим извлечением структуры и содержания. Подход может терять семантическую информацию, но проще в реализации.

***Г***. *Гибридные решения* - комбинация различных подходов, например, использование парсера для базовой структуры документа и переопределение макрокоманд для специфических конструкций.

***Д***. *Использование нейросетевых моделей* - использование нейросетевых моделей для трансформации исходного текста LaTeX в XML. Этот подход позволяет достичь высокой точности конвертации, но требует значительных вычислительных ресурсов и большого объема данных для обучения модели.

Самый качественный и универсальный подход - использование нейросетевых моделей, но высокие требования к вычислительным ресурсам и объему данных для обучения модели усложняют использование этого подхода.

Использование промежуточных форматов позволяет частично упростить обработку, однако сталкивается с проблемами при работе со старыми и неподдерживаемыми форматами документами. Также создаются накладные расходы на хранение и обработку промежуточных форматов.

Рассмотрены *методы и инструменты* преобразования документов по извлечению структурированной информации [20, 21] из неструктурированных или полуструктурированных текстов, что имеет прямую аналогию с задачей преобразования LaTeX-разметки в семантически обогащенный XML.

*Универсальные конвертеры*: Проанализированы возможности таких инструментов, как Pandoc и TeX4ht (На данный момент репозиторий TeX4ht использует доменные имена, заблокированные в РФ, что существенно затрудняет его использование и внедрение.)

*Программатические подходы (programmatic approach)*: Работы по созданию *кастомных парсеров* для LaTeX с использованием регулярных выражений, синтаксических анализаторов (например, ANTLR) или специализированных библиотек для работы с текстовыми данными. Это позволило оценить сложность разработки собственного решения и его преимущества в плане точности и контроля над выходным форматом.

Наиболее простым и надежным подходом представляется *гибридный*: *комбинация парсинга для базовой структуры документа и переопределения макрокоманд для специфических конструкций и пользовательских команд*.

### 3.2. *Разработка архитектуры преобразователя исходного документа и построение соответствующего XML-документа*

Разработка системы конвертации LaTeX в XML реализована путем создания лексера (лексического анализатора) и парсера (синтаксического анализатора), авторского гибридного «компилятора» [22, 23, 24].

Для начала был проделан эксперимент, чтобы избежать самостоятельного написания лексера и парсера при условии универсальности их применения была произведена тестовая сборка файлов с помощью существующих систем сборки LaTeX.

***Сборка файлов***. Несмотря на наличие собранных в архиве PDF-файлов, было принято решение провести тестовую сборку, для проверки возможности использования решения, основанного на генерации промежуточного файла.

В ходе сборки были выявлены проблемы (**П0.- П1.**):

**П0. Кодировка**.

Кодировка файлов произведена в CP866 (также известна, как IBM866). CP866 – «альтернативная кодировка», основана на CP437, отличается тем, что большинство специфических европейских символов в верхней половине (позициях 0x80 - 0xFF) кодовой таблицы заменено на буквы русского алфавита, а псевдографические символы оставлены нетронутыми.

Кодировка CP866 разработана в паре с основной кодировкой (с которой совпадает по набору символов) в середине 1980-х годов в Вычислительном центре Академии наук СССР [25]. Эта кодировка

пользовалась большой популярностью среди советских пользователей IBM PC-совместимых ПК.

Все кириллические символы нечитаемы в исходном виде, в связи с чем, встала задача перекодировки файлов в кодировку UTF-8, как основную систему кодировки документов для подавляющего большинства повседневных задач для текстов в LaTeX.

Проблема **П0.** была решена. Для преобразования файлов написан скрипт на языке Python, изменяющий кодировку с сохранением исходного файла с расширением «.backup». В ходе работы выяснилось, что существующие механизмы обнаружения кодировок (разные версии uchardet) ***не*** определяют CP866 корректно (во всех случаях кодировка определена как турецкий подтип ISO). Это связано с тем, что CP866 — единственная или одна из немногих IBM-образных таблиц, получивших достаточно широкое распространение и при этом *не вошедших в спецификацию* ISO.

**П1. Специфические библиотеки**

Некоторые библиотеки (в частности, «acad.sty») не содержатся в основных репозиториях и пакетах поставки LaTeX-систем на Linux-дистрибутивы, а CTAN более не считает пакет активным.

Проблема **П1.** не решена. Предполагается возможность подбора требуемой версии и параметров сборки библиотеки ручным способом, однако количество времени и ресурсов, которые необходимо для этого затратить нецелесообразно в сравнении с предполагаемым полезным результатом.

В результате эксперимента с тестовой сборкой файлов было выработано решение, разработки собственного лексера-парсера (компилятора) и последующий траверс AST (Abstract Syntax Tree).

1. Вариант с использованием нейросетей не рассматривался как излишне время-, ресурсо- и трудозатратный.
2. Вариант с использованием промежуточного представления требовал компилируемости сборки, что не удалось (**П1**).
3. Автоматное решение само по себе не актуально и не соответствует масштабу поставленной задачи.

Выбраны существующие генераторы грамматик (ANTL4 + Python). Разделение на лексер и парсер не осуществлялось, единая грамматика была сформирована с целью упрощения организации правил и управления конфигурацией.

### 3.3. *Реализация прототипа конвертера, удовлетворяющего заявленным требованиям*.

В числе предоставленных для работы файлов был пример формата XML, конвертацию в который требовалось осуществить. С целью

облегчения и ускорения отладки были составлены *модельные примеры корректных входного и выходного файла*.

В процессе анализа исходных файлов для построения модельного, выяснилось следующие проблемы (**П2.-П4.**):

**П2. Пользовательские макрокоманды**

Все (или почти все) файлы с «полезным содержимым», то есть, декларативным LaTeX-кодом, представленные в предложенном примере, содержат определения пользовательских макрокоманд, нередко встречается переопределение, замещение одной макрокоманды другой, при этом не всегда с сохранением смысла. В самом файле IPI.TEX количество и содержание переопределений также нестабильно и меняется от выпуска к выпуску. Вероятнее всего, это связано с различной тематикой выпусков (например, определение *D* для быстрого начертания *дисперсии* для выпуска, посвящённого теории вероятностей).

Данная проблема существенно осложняет разработку компиляторного решения и в очередной раз указывает на удобство нативного (оптимизированного для конкретного случая) решения (использования существующих систем сборки для генерации промежуточного или конечного файла и последующей обработки такового).

Проблема **П2.** решена. Для устранения был написан Python-скрипт, который в два прохода по файлам производит раскрытие пользовательских макросов (запись-раскрытие: 2 прохода). При этом пользовательские макросы (такие, которые определены в файлах статей) имеют приоритет над глобальными. Данное действие относится к предобработке данных и выполняется до запуска непосредственного решения, аналогично с программой смены кодировки.

Написанный прототип *успешно конвертировал модельный исходный файл в модельный целевой файл*, опираясь на очевидные семантические маркеры (например, форматированная строка «Доказательство» для отделения доказательства присутствовала во всех файлах в том или ином виде.)

Грамматика для модельного файла занимала 107 строк без учета сопроводительных пометок.

В ходе реализации грамматики для «рабочего», а не модельного файла, объём грамматики быстро вырос до ~400 строк, а ANTLR-генерированные Python-программы стали нестабильны. Появились ошибки, например разбор текстового сегмента как правила после перевода на новую строку (символ абзаца в LaTeX - «\\», при этом LaTeX-правила начинаются с «\»).

**П3. ANTLR4 и сложные грамматики**

В ходе работы над практической грамматикой стало очевидно, что фреймворк ANTLR4 представляет собой мощный и расширяемый генератор парсеров по заданным правилам. Однако конфигурация ANTLR4 строго

формализована и *не допускает ручного управления и временного отхода от строгих правил* реализации грамматических правил обработки языка, что осложняет разработку практических лексеров и парсеров. В частности, при реализации рекурсивного спуска вручную, возможно реализовать «наивный lookahead» путем поиска закрывающего тэга с углублением рекурсии при обнаружении повторного открывающего. С ANTLR4, как и с любым другим генератором это невозможно.

**П4. ANTLR4 и правила**

Несмотря на отличную работу генератора на модельных файлах, попытка применить его на практике столкнулась с *проблемой конфликтующих правил*, а также нетривиальной приоритетной очередности составленных выражений. К моменту, когда грамматика для практического файла достигла ~400 строк и ее отладка стала проблематичной, стало очевидно, что использование функционала грамматических генераторов необходимо пересмотреть.

В результате процесса расширения прототипа стало очевидно, что использование ANTL4 сопряжено с необходимостью считаться с особенностями подхода. Так как поддержка трудноотлаживаемой и слабоуправляемой системы осложняет применение и внедрение решения, внимание было обращено на существующие решения лексинга и парсинга.

### *3.4. Оценка эффективности предложенного подхода путем экспериментального тестирования и сравнения результатов с существующими аналогичными инструментами*

Траверс существующего дерева

*Постановка*: В связи с исключительной сложностью LaTeX, имеет смысл использовать существующую систему анализа языка для формирования AST, по которому производить траверс, то есть, воспользоваться системой какого-либо существующего лексера и парсера LaTeX для более корректного и качественного разбора текста.

Были рассмотрены основные популярные «движки»: pdftex, xetex, luatex, bibtex, miktex. В ходе анализа выяснилось, что как такового, построения AST они не производят.

**П5. LaTeX-компиляция**

Все основные системы компоновки LaTeX в визуально читаемые форматы вместо лексинга и парсинга, де-факто раскрывают встреченные макросы заранее или динамически определенным образом. Поэтому, использовать существующее абстрактное синтаксическое дерево не получится из-за его отсутствия.

В связи с выяснением данного обстоятельства, принято решение опробовать другие системы LaTeXML и плоский LaTeX.

**LaTeXML.** Мощная система, написанная на Perl, предназначенная для преобразования документов LaTeX в различные XML-форматы, включая XHTML, MathML, DocBook и другие. LaTeXML создана энтузиастами и применена в проекте arXiv для генерации HTML-страниц из массивной библиотеки накопленной научной литературы. LaTeXML имеет расширяемую систему конфигурации, вероятностно-эвристические механизмы решения конфликтов и ошибок, а также поддерживает спецификацию требуемого выходного формата в виде XML-шаблона.

**П6. LaTeXML**

Основными недостатками оказались: неспособность использовать семантические маркеры (не-LaTeX команды) в качестве управляющих символов для правил без вмешательства в систему разбора исходных файлов; высокий порог вхождения для корректной настройки шаблонов и собственных макросов; требование к исходному файлу должно быть представленным в виде отдельно стоящего, не имеющего зависимостей (есть поддержка зависимостей, но в рамках задачи она не имеет смысла, т. к. требуется скорее обратный разбор файла — подключение импортируемого файла в импортирующий файл во избежание слияние статей).

После нескольких попыток построить конфигурацию с учетом семантических маркеров и зависимостей, было решено отказаться от LaTeXML как от системы, не подходящей для данной задачи (**П6.**). Стало необходимо разрешить проблемы организации импортов и пользовательских макросов. Именно на этом этапе была реализована рабочая *система раскрытия большинства макросов методом двойного прохода по файлу*

**Плоский LaTeX**. Идея плоского LaTeX заключается в следующем: чтобы избежать сложностей в виде разрешения зависимостей и раскрытия пользовательских макрокоманд, можно сделать обратное — слить все статьи в единый выходной LaTeX-файл, в котором раскрыть все макрокоманды с учетом приоритета и области видимости.

Реализация алгоритма состояла из двух подпрограмм:

1. Программа раскрытия макросов
2. Программа разрешения подключений зависимостей и объединения текста

1. На первом проходе алгоритм фокусируется на идентификации и извлечении определений макрокоманд. В рамках данной фазы Python-скрипт осуществляет итеративное считывание исходного LaTeX-документа (построчно или блоками). Применяются специализированные регулярные выражения для обнаружения синтаксических конструкций, таких как \newcommand, \renewcommand, \def, а также \newenvironment. При успешной валидации такого определения производится сохранение имени макроса (например, \myMacro) и его ассоциированного тела (строковой репрезентации, на которую должен быть произведен замещающий процесс).

Эти пары «ключ-значение» (имя макроса: тело) хранятся в словаре (используется Python). На данном этапе непосредственная модификация исходного файла не осуществляется; происходит исключительно аккумулирование макросов.

2. Второй проход предназначен для непосредственного раскрытия макрокоманд. Алгоритм осуществляет повторное считывание (или оперирует с уже загруженным в оперативную память контентом) LaTeX-файла. Для каждого сегмента текста, применяются регулярные выражения, сконфигурированные для поиска вызовов (инвокаций, выполнении последовательности команд, записанных в макросе) макросов (например, \myMacro или \myMacro{аргумент}). При обнаружении инвокации макроса его идентификатор верифицируется в словаре, сформированном на предыдущем проходе. Далее, с использованием функционала re.sub() библиотеки Python, обнаруженное использование макроса замещается его соответствующим телом из словаря. В случае, если макрос предполагает наличие аргументов, регулярное выражение должно быть сконструировано с достаточной сложностью для захвата этих аргументов и их последующей параметризации в теле макроса до осуществления операции замещения. Данный процесс *итеративно повторяется* до момента раскрытия всех известных макросов. Если определение макроса неявно содержит другие макросы, данный проход может потребовать циклических замещений до полного раскрытия всех вложенных структур, что обеспечивает детерминированность конечного результата.

**П7. Отсутствие разграничения**

Несмотря на то, что раскрытие макросов было призвано повысить легкость разбора LaTeX, обнаружилось, что некоторые макрокоманды удобнее парсить ***до*** раскрытия. Например, макросы заголовка статьи, авторов, ключевых слов и т. д. гораздо удобнее обрабатывать до раскрытия, т. к. после раскрытия одно слово макроса может превратиться в сложную систему LaTeX-команд, не представляющих интереса для разбора.

В результате, в последовательность действий была внесена корректировка, а именно: *программа до первого прохода считывает список «запретных» макрокоманд из указанного файла и игнорирует их при раскрытии*.

После получения нескольких экземпляров плоского LaTeX стало очевидно, что ситуация улучшилась, однако обрабатывать такой файл стало сложнее (**П7, П8.**).

**П8. Большие плоскости**

Очевидно, что с ростом размера файла, легче его обрабатывать не становится (в большинстве случаев). Обработка полного текста выпуска журнала оказалась неверным решением по ряду причин:

А). в качестве результата работы ожидались раздельные файлы статей, а не монолитный XML-журнал;

Б). с ростом размера файла увеличивается необходимый объем ресурсов вычислительной системы, время обработки, осложняется обработка и нивелирование ошибок.

В связи с этим необходимо модифицировать набор программ: *вместо объединения файлов в единый монолитный плоский LaTeX, каждая статься приводится к плоскому LaTeX индивидуально и обрабатывается авторским лексером и парсером.*

При доработке данного решения рекомендуется провести работу над уточнением возможных вариаций однотипных конструкций (например, разных видов определения математических формул: «\equation», «»$, «$$», «\[\]», «\(\)», …).

Также имеет смысл произвести расчетную оценку абстрактной точности решения, так как на данный момент в ходе трансляции ошибочные места могут приводить к бесконечным циклам и критическим ошибкам.

Все перечисленные проблемы учитываются в процессе реализации предобработки и загрузки новых данных в библиотеку.

## 4. Заключение и выводы

В результате исследования была проведена оценка качества данных библиотеки *SciLibRu,* выявлены направления для их совершенствования, как на этапе загрузки (предобработки данных с помощью гибридного подхода) так и на этапе достраивания онтологии и *KG.* Приведены примеры интеграции *LLM* с *KG SciLibRu* для коммуникации на естественном языке. Получен набор структурированных данных, который может быть использован для обучения *LLM* на корпусе математических русскоязычных научных статей.

Дальнейшие исследования направлены на улучшение качества данных *SciLibRu* с целью формирования *датасета* для работы *LLM* с русскоязычными научными текстами.

Работа представлена в рамках выполнения темы НИР «Математические методы анализа данных и прогнозирования» ФИЦ ИУ РАН.

### Литература

1. Серебряков В.А., Атаева О.М. Информационная модель открытой персональной семантической библиотеки LibMeta // Научный сервис в сети Интернет: труды XVIII Всероссийской научной конференции (19-24 сентября 2016 г., г. Новороссийск). М.: ИПМ им. М.В.Келдыша, 2016. С. 304-313. URL: http://keldysh.ru/abrau/2016/3.pdf.
2. Rospocher M., Tonelli S., Serafini L.and Pianta E. Corpus-based terminological evaluation of ontologies // Applied Ontology 7(4) (2012), 429–448.
3. Ataeva O., Serebryakov V., and Tuchkova N., Ontological approach to a knowledge graph construction in a semantic library // Lobachevskii J. of

Mathematics, 44, (6) (2023) pp. 2229–2239. https://doi.org/10.1134/S1995080223060471.
4. Handbook on Ontologies. Editors: Steffen Staab Rudi Studer, Springer-Verlag Berlin Heide1berg 2004 DOI 10.1007/978-3-540-24750-0.
5. Атаева О.М., Серебряков В.А., Тучкова Н.П. Подходы к организации математических знаний при формирования предметных тезаурусов различных разделов математики // CEUR Workshop Proceedings. 2018. Vol. 2260. P. 42-54. ISSN:1613-0073. https://doi.org/10.20948/abrau-2018-66.
6. Hlomani H., Stacey D., Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey //Semantic Web Journal, 2014, 1(5) pp. 1–11.
7. Lozano-Tello A. and Gómez-Pérez A. Ontometric: A method to choose the appropriate ontology // Journal of Database Management (JDM) 15(2) (2004), 1–18.
8. Шрейдер Ю.А. Тезаурусы в информатике и теоретической семантике // Научно-техническая информация. Сер. 2. 1971. № 3. С. 21–24.
9. Лукашевич, Н.В. Тезаурусы в задачах информационного поиска. М.: Изд-во МГУ, 2011. 495 с.
10. Харари Ф. Теория графов. Пер. с англ. и предисл. В. П. Козырева. Под ред. Г.П.Гаврилова. Изд. 2-е. ‘ М.: Едиториал УРСС, 2003. 296 с.
11. Barrasa J., Webber J., Building Knowledge Graphs: A Practitioner’s Guide. O'Reilly. 2023. 290 p.
12. Biswas, G., Bezdek, J., and Oakman, R. L.: A knowledge-based approach to online document retrieval system design. In Proc. ACM SIGART Int. Symp. Methodol. pp. 112–120. Intell. Syst. 1986.
13. Гаврилова Т.А., Кудрявцев Д.В., Муромцев Д.И. Инженерия знаний. Модели и методы: Учебник. СПб.: Издательство «Лань», 2016. 324 с.
14. Pan S., at all. Unifying Large Language Models and Knowledge Graphs: A Roadmap // in *IEEE Transactions on Knowledge and Data Engineering/* Pan S., Luo L., Wang Y., Chen C., Wang J., Wu X., V.. 36. №. 7. P. 3580-3599, July 2024, https://doi.org/10.1109/TKDE.2024.3352100.
15. Luo L., at all. Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models. / Luo L., Zhao Z., Gong C., Haffari G., Pan S. arXiv preprint arXiv:2410.13080. 2024. https://doi.org/10.48550/arXiv.2410.13080.
16. Виноградов И.М. (Гл. ред.) “Математическая энциклопедия” (в 5 томах) М.: Советская энциклопедия (1977—1985).
17. Фаддеев Л. Д. (Гл. ред.) “Энциклопедия математической физики. Энциклопедия. М.: Большая русская энциклопедия.1998. 692 с.

18. Ataeva O.M., Tuchkova N.P. Adaptation of the language model for mathematical texts in the semantic library// System Informatics (Системная информатика), 2025, No.27, pp. 59-75.
19. Будзко В.И., Атаева О.М., Тучкова Н.П. Автоматизация доступа к информации при навигации по данным семантической библиотеки и интеграции графа знаний с языковой моделью // Системы высокой доступности. 2025. Т. 21. № 2. С. 5−11. https://doi.org/10.18127/j20729472-202502-0.
20. Клюкин А.А., Широков А.А. Автоматизированная система подготовки слабоструктурированной информации / А. А. Клюкин, Широков [Электронный ресурс] // Гаудеамус. 2014. №2 (24). URL: https://cyberleninka.ru/article/n/avtomatizirovannaya-sistema-podgotovki-slabostrukturirovannoy-informatsii (дата обращения: 10.06.2025).
21. Куртюкин С.В. Метод автоматизированного формирования сборников архивных документов [Электронный ресурс] // Теория и практика современной науки. 2018. №5 (35). URL: https://cyberleninka.ru/article/n/metod-avtomatizirovannogo-formirovaniya-sbornikov-arhivnyh-dokumentov (дата обращения: 17.06.2025).
22. Ахо А. Компиляторы: принципы, технологии, инструменты / А. Ахо, Р. Сети, Дж. Ульман. Москва : Вильямс, 2001.
23. Волкова И.А., Вылиток А.А., Руденко Т.В.. Формальные грамматики и языки. Элементы теории трансляции: учебное пособие для студентов II курса Москва : Изд-во Моск. гос. ун-та, 2009.
24. Гладкий А. В. Формальные грамматики и языки. Москва : Наука, Гл. ред. физ.-мат. лит., 1973.
25. Брябрин В. М., Ландау И. Я., Неменман М. Е. О системе кодирования для персональных ЭВМ // Микропроцессорные средства и системы. 1986. № 4. С. 61–64.

**References**

1. Serebryakov V.A., Ataeva O.M. Informacionnaya model' otkrytoj personal'noj semanticheskoj biblioteki LibMeta // Nauchnyj servis v seti Internet: trudy XVIII Vserossijskoj nauchnoj konferencii (19-24 sentyabrya 2016 g., g. Novorossijsk). M.: IPM im. M.V.Keldysha, 2016. S. 304-313.URL: http://keldysh.ru/abrau/2016/3.pdf.
2. Rospocher M., Tonelli S., Serafini L.and Pianta E. Corpus-based terminological evaluation of ontologies // Applied Ontology 7(4) (2012), 429–448.
3. Ataeva O., Serebryakov V., Tuchkova N., Ontological approach to a knowledge graph construction in a semantic library // Lobachevskii J. of

Mathematics, 44, (6) (2023) pp. 2229–2239. https://doi.org/10.1134/S1995080223060471.
4. Handbook on Ontologies. Editors: Steffen Staab Rudi Studer, Springer-Verlag Berlin Heide1berg 2004 DOI 10.1007/978-3-540-24750-0.
5. Ataeva O., Serebryakov V., Tuchkova N., Podhody k organizacii matematicheskih znanij pri formirovaniya predmetnyh tezaurusov razlichnyh razdelov matematiki // CEUR Workshop Proceedings. 2018. Vol. 2260. P. 42-54. ISSN:1613-0073. https://doi.org/10.20948/abrau-2018-66.
6. Hlomani H., Stacey D., "Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey" Semantic Web Journal, 2014, 1(5) pp. 1–11.
7. Lozano-Tello A. and Gómez-Pérez A. Ontometric: A method to choose the appropriate ontology // Journal of Database Management (JDM) 15(2) (2004), 1–18.
8. Shrejder YU.A. Tezaurusy v informatike i teoreticheskoj semantike // Nauchno-tekhnicheskaya informaciya. Ser. 2. 1971. № Z. S. 21–24.
9. Lukashevich, N.V. Tezaurusy v zadachah informacionnogo poiska. M.: Izd-vo MGU, 2011. 495 s.
10. Harari F. Teoriya grafov. Per. s angl. i predisl. V. P. Kozyreva. Pod red. G.P.Gavrilova. Izd. 2-e. ' M.: Editorial URSS, 2003. 296 s.
11. Barrasa J., Webber J., Building Knowledge Graphs: A Practitioner's Guide. O'Reilly. 2023. 290 p.
12. Biswas, G., Bezdek, J., and Oakman, R. L.: A knowledge-based approach to online document retrieval system design. In Proc. ACM SIGART Int. Symp. Methodol. pp. 112–120. Intell. Syst. 1986.
13. Gavrilova T.A., Kudryavcev D.V., Muromcev D.I. Inzheneriya znanij. Modeli i metody: Uchebnik. SPb.: Izdatel'stvo «Lan'», 2016. 324 s.
14. Pan S., at all. Unifying Large Language Models and Knowledge Graphs: A Roadmap // in *IEEE Transactions on Knowledge and Data Engineering/* Pan S., Luo L., Wang Y., Chen C., Wang J., Wu X., V.. 36. №. 7. P. 3580-3599, July 2024, https://doi.org/10.1109/TKDE.2024.3352100.
15. Luo L., at all. Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models. / Luo L., Zhao Z., Gong C., Haffari G., Pan S. arXiv preprint arXiv:2410.13080. 2024. https://doi.org/10.48550/arXiv.2410.13080.
16. Vinogradov I.M. (Gl. red.) "Matematicheskaya enciklopediya" (v 5 tomah) M.: Sovetskaya enciklopediya (1977—1985).
17. Faddeev L. D. (Gl. red.) "Enciklopediya matematicheskoj fiziki. Enciklopediya. M.: Bol'shaya russkaya enciklopediya.1998. 692 s.

18. Ataeva O.M., Tuchkova N.P. Adaptation of the language model for mathematical texts in the semantic library// System Informatics, 2025, No.27, pp. 59-75.
19. Budzko V.I., Ataeva O.M., Tuchkova N.P. Access automation to information for navigating through semantic library data and integrat-ing the knowledge graph with the language model. Highly Available Systems. 2025. V. 21. № 2. P. 5−11. DOI: https://doi.org/ 10.18127/j20729472-202502-0 (in Russian)
20. Klyukin A.A., SHirokov A.A. Avtomatizirovannaya sistema podgotovki slabostrukturirovannoj informacii / A. A. Klyukin, SHirokov [Elektronnyj resurs] // Gaudeamus. 2014. №2 (24). URL: https://cyberleninka.ru/article/n/avtomatizirovannaya-sistema-podgotovki-slabostrukturirovannoy-informatsii.
21. Kurtyukin S.V. Metod avtomatizirovannogo formirovaniya sbornikov arhivnyh dokumentov [Elektronnyj resurs] // Teoriya i praktika sovremennoj nauki. 2018. №5 (35). URL: https://cyberleninka.ru/article/n/metod-avtomatizirovannogo-formirovaniya-sbornikov-arhivnyh-dokumentov.
22. Aho A. Kompilyatory: principy, tekhnologii, instrumenty / A. Aho, R. Seti, Dzh. Ul'man. Moskva : Vil'yams, 2001.
23. Volkova I.A., Vylitok A.A., Rudenko T.V.. Formal'nye grammatiki i yazyki. Elementy teorii translyacii : uchebnoe posobie dlya studentov II kursa Moskva : Izd-vo Mosk. gos. un-ta,2009.
24. Gladkij A. V. Formal'nye grammatiki i yazyki. Moskva : Nauka, Gl. red. fiz.-mat. lit., 1973.
25. Bryabrin V. M., Landau I. YA., Nemenman M. E. O sisteme kodirovaniya dlya personal'nyh EVM // Mikroprocessornye sredstva i sistemy. 1986. № 4. S. 61–64.