

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Кафедра дискретной математики и алгоритмики

ГОРБАЧ Екатерина Витальевна

**ИСПОЛЬЗОВАНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ
ДЛЯ ИЗВЛЕЧЕНИЯ ИНФОРМАЦИИ ИЗ ТАБЛИЧНЫХ ДАННЫХ**

Магистерская диссертация
специальность 1-31 80 09 Прикладная математика и информатика

Научный руководитель
Соболевская Елена Павловна
кандидат физико-математических наук,
доцент

Допущена к защите:
«30» марта 2023 г
Зав. кафедрой дискретной математики и алгоритмики
Котов Владимир Михайлович
доктор физико-математических наук, профессор

Минск, 2023

ОГЛАВЛЕНИЕ

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ.....	4
ВВЕДЕНИЕ.....	5
ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ.....	7
ГЛАВА 1 ЗАДАЧА ИЗВЛЕЧЕНИЯ ИНФОРМАЦИИ ИЗ ТАБЛИЦ И МЕТОДЫ ЕЕ РЕШЕНИЯ.....	10
1.1 Задача извлечения информации из таблиц.....	10
1.2. Смежные задачи.....	11
1.2.1 Задачи извлечения информации из таблицы по Zang и Balog.....	11
1.2.2 Задачи извлечения информации из таблицы по Jan C и др.....	13
1.2.3 Задача понимания таблиц по Braunschweig.....	14
1.3 Таксономия таблиц.....	15
1.3.1 Анатомия таблицы.....	15
1.3.2 Развитие таксономий таблиц.....	16
1.4 Коллекции таблиц для задачи определения класса таблицы.....	19
1.5 Обзор методов решения смежных задач.....	19
1.6 Значимость задачи исследования.....	22
1.7 Выводы.....	23
ГЛАВА 2 РЕШЕНИЕ ЗАДАЧИ.....	25
2.1 Выбранные методы решения задачи выделения структуры в таблицах... 26	26
2.1.1 Эвристика в задаче определения класса таблицы.....	27
2.1.2 Методы машинного обучения в задаче определения класса таблицы... 27	27
2.1.3 Методы глубокого обучения в задаче определения класса таблицы. 30	30
2.1.4 Методы машинного обучения в задаче определения роли ячеек в таблице.....	32
2.2 Алгоритм выделения кортежей из таблицы.....	39
ГЛАВА 3 ПРИМЕНЕНИЕ МЕТОДОВ ОПРЕДЕЛЕНИЯ КЛАССА ТАБЛИЦЫ.....	41
3.1 Состав данных для обучения и сравнения моделей.....	41
3.2 Подготовка данных.....	42
3.3 Применение эвристики.....	42
3.4 Применение градиентного бустинга.....	43
3.4.1 Признаковое описание таблицы.....	43
3.4.2 Обучение модели и полученные результаты.....	43
3.5 Применение нейронных сетей.....	44
3.6 Выводы.....	46

ГЛАВА 4 ПРИМЕНЕНИЕ МЕТОДОВ ОПРЕДЕЛЕНИЯ КЛАССА	
ЯЧЕЕК.....	48
4.1 Состав данных для обучения и сравнения моделей.....	48
4.2 Подготовка данных.....	49
4.3 Применение нейронных сетей.....	50
4.3.1 Transfer learning: обучение на коллекциях DeEX, SEUS.....	50
4.3.2 Обучение на коллекции таблиц TaxDataset.....	51
4.4 Выводы.....	52
ЗАКЛЮЧЕНИЕ.....	54
ПРИЛОЖЕНИЕ А.....	55
ПРИЛОЖЕНИЕ Б.....	56
ПРИЛОЖЕНИЕ В.....	57
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	58

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

1. **Векторное представление (эмбединг, embedding)** — это вектор фиксированной длины, который получается из вектора большей размерности (соответствующего каким-то объектам, например, картинкам или словам) при помощи заранее определенного преобразования. При этом это преобразование обладает тем свойством, что семантически похожие объекты отображаются в близкие векторы в пространстве эмбедингов.
2. **Семантическое векторное представление ячейки (semantic cell embedding)** — векторное представление ячейки таблицы, которое использует семантические свойства текста, содержащегося в ячейке.
3. **Стилистическое векторное представление ячейки (style cell embedding)** — векторное представление ячейки таблицы, которое использует стиль текста ячейки.
4. **LSTM (Long Short-Term Memory)** — это вид рекуррентной нейронной сети, способной обрабатывать и моделировать последовательности данных, учитывая долгосрочные зависимости в этой последовательности. Использует специальный механизм памяти, состоящий из трех "вентилей" (входной, выходной и забывающий).
5. **Softmax** — это функция активации, которая используется в нейронных сетях для преобразования вектора чисел в вероятностное распределение.
6. **Автоэнкодер (auto encoder)** — это нейронная сеть, которая учится кодировать и декодировать данные. Данные, которые подаются на вход автоэнкодеру, сжимаются в более компактное представление, которое называют векторным представлением или embedding-ом. После процесса сжатия автоэнкодер пытается восстановить исходные данные из этого векторного представления.

ВВЕДЕНИЕ

В настоящее время в открытом доступе хранится множество информации разной структуры, которую человек своими силами обработать и изучить не может, поэтому инструменты, помогающие интерпретировать и хранить данные в более компактном виде, становятся все более необходимы и полезны.

Таблицы являются распространенным инструментом отображения информации, так как человек может быстро и наглядно их интерпретировать. По некоторым оценкам [1] сейчас в Интернете существуют сотни миллионов таблиц. В [2] также отметили взрыв потребительского спроса на данные, поступающие из таблиц, благодаря растущей потребности уметь совершать поисковые запросы не только над текстовыми данными, но и над данными более сложной структуры.

Приложения, использующие информацию, извлеченную из таблиц, могут помочь улучшить процесс принятия решений, автоматизировать задачи и улучшить взаимодействие с пользователем. KB-augmentation — одно из таких приложений, которое включает в себя дополнение базы знаний информацией, извлеченной из таблиц. Это может помочь повысить точность и полноту базы знаний, сделав ее более полезной для различных приложений, таких как чат-боты и виртуальные помощники. Поиск по таблице — еще одно приложение, которое может быть полезно для быстрого поиска релевантной информации в больших наборах данных. Вопросно-ответные системы (QA-системы) — еще один пример приложений, использующих информацию из таблиц. Эти системы могут отвечать на вопросы на естественном языке, извлекая информацию из таблиц и других структурированных источников данных, что делает их полезными для различных приложений, таких как поддержка клиентов и анализ данных. В целом, приложения, использующие информацию, извлеченную из таблиц, могут повысить эффективность, точность и удобство работы пользователей в различных отраслях.

Автоматическая обработка таблиц также может быть полезна при сборе и хранении данных, которые традиционно представлены в полуструктурированном виде: квитанции, чеки. Много данных такого формата не имеют однотипную структуру и нуждаются в предобработке для последующего анализа и хранения.

Сложности, которые возникают при обработке таблиц, связаны с тем, что таблицы содержат данные на естественном языке, а также имеют структуру, которую необходимо уметь определять. На данный момент нет готового решения, которое смогло бы определить структуру любой семантически

правильной таблицы. При этом для человека эта задача ясна и интуитивно понятна.

Существующие решения концентрируются на интерпретации простых таблиц и не затрагивают многие виды таблиц, например, таблицы с иерархическими заголовками. Также коллекции таблиц, по которым измеряется качество новых алгоритмов, имеют узкую специализацию и не покрывают многие жизненные задачи.

В рамках магистерской диссертации исследуется задача извлечения информации из таблиц, полученных из источников, связанных с налоговой деятельностью и методы ее решения, основанные на машинном обучении. Такого рода таблицы имеют сложную структуру и ряд особенностей, не представленных в коллекциях таблиц, используемых в научной литературе.

В главе 1 магистерской диссертации рассматриваются различные аспекты, связанные с извлечением информации из таблиц, приводятся различные постановки задач для этой области и вводятся основные определения. Более того, в главе 1 показывается, что методы машинного обучения являются наиболее распространенными и эффективными способами автоматической обработки таблиц.

На основании изложенного в главе 1, в главе 2 формулируется постановка задачи, исследуемой в магистерской диссертации. Задача декомпозируется на два этапа и для первого этапа предлагаются решения для таблиц простой и сложной структуры.

В рамках глав 3-4 излагаются результаты экспериментов, а также приводится описание модификаций выбранных методов решения.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Магистерская диссертация, 59 страниц, 28 рисунков, 18 таблиц, 22 источника, 3 приложения.

АВТОМАТИЧЕСКАЯ ОБРАБОТКА ДОКУМЕНТОВ, ВЕБ-ТАБЛИЦЫ, РАСПОЗНАВАНИЕ СТРУКТУРЫ ВЕБ-ТАБЛИЦ, МАШИННОЕ ОБУЧЕНИЕ, НЕЙРОННЫЕ СЕТИ

Объект исследования: применение методов машинного обучения для решения задачи извлечения информации из таблиц, полученных из источников, связанных с налоговой деятельностью.

Цель работы: изучение и анализ существующих алгоритмов машинного обучения; разработка и реализация алгоритма обучения модели нейронной сети для рассматриваемой задачи; сравнительный анализ полученных результатов; определение возможных направлений по улучшению предложенного метода решения задачи, исследуемой в работе.

Методы исследования: сбор и подготовка данных для обучения и тестирования моделей нейронных сетей; построение нейронной архитектуры и ее оптимизация с учетом особенностей предметной области и данных.

Область применения: автоматический сбор и хранение данных, представленных в виде веб-таблиц сложной структуры, поиск по таблицам.

АГУЛЬНАЯ ХАРАКТАРЫСТЫКА ПРАЦЫ

Магістарская дысертацыя, 59 старонак, 28 малюнкаў, 18 табліц, 22 крыніцы, 3 дадатка.

Аўтаматычная апрацоўка дакументаў, вэб-табліцы, распазнаванне структуры вэб-табліц, машыннае навучанне, нейронныя сеткі.

Аб'ект даследавання: прымяненне метадаў машыннага навучання для рашэння задачы вынятку інфармацыі з табліц, атрыманых з крыніц, звязаных з падатковай дзейнасцю.

Мэта работы: вывучэнне і аналіз існуючых алгарытмаў машыннага навучання; распрацоўка і рэалізацыя алгарытма навучання мадэлі нейронавай сеткі для разгляданай задачы; параўнальны аналіз атрыманых вынікаў; вызначэнне магчымых напрамкаў па палепшанні прапанаванага метада рашэння задачы, даследаванай у працы.

Метады даследавання: збор і падрыхтоўка дадзеных для навучання і тэставання мадэляў нейронных сетак; пабудова нейронавай архітэктурны і яе аптымізацыя з улікам асаблівасцей прадметнай галіны і дадзеных.

Вобласць прымянення: аўтаматычны збор і захоўванне дадзеных, прадстаўленых у выглядзе вэб-табліц складанай структуры, пошук па табліцах.

GENERAL DESCRIPTION OF WORK

Master's thesis, 59 pages, 28 figures, 18 tables, 22 sources, 3 appendix.

AUTOMATIC DOCUMENT PROCESSING, WEB TABLES, TABLE STRUCTURE RECOGNITION, MACHINE LEARNING, NEURAL NETWORKS.

The object of research: application of machine learning algorithms to solve the task of information retrieval from table documents obtained from sources related to tax activity.

The aim of this work: research and analysis of existing machine learning algorithms; development and implementation of a neural network model algorithm for the considered task; comparative analysis of the obtained results and identification of possible directions for improving the proposed method for solving the problem under study.

Research methods: data collection and preparation for training and testing neural network models; developing the neural architecture and fine-tuning it to match the unique features of the subject domain and the dataset.

Field of application: automatic collection and storage of data presented in the form of complex-structured web tables; table search.

ГЛАВА 1

ЗАДАЧА ИЗВЛЕЧЕНИЯ ИНФОРМАЦИИ ИЗ ТАБЛИЦ И МЕТОДЫ ЕЕ РЕШЕНИЯ

1.1 Задача извлечения информации из таблиц

Таблицы — это мощный и популярный инструмент для организации данных и управления ими. В Интернете можно найти огромное количество таблиц, которые представляют собой ценный ресурс знаний. В частности, квитанции и чеки зачастую имеют сложную структуру и представлены в виде таблиц (рисунок 1.1). В открытом доступе легко найти данные по налогообложению частной собственности некоторых стран. Источником таблиц могут быть pdf-документы, excel-таблицы, веб-таблицы и др. В данном исследовании мы ограничимся веб-таблицами, так как это наиболее доступный источник информации. Результаты, полученные на веб-таблицах, при ряде преобразований могут быть перенесены на таблицы других форматов.

CURRENT STATUS AND SUMMARY			
	FIRST INSTALLMENT	SECOND INSTALLMENT	TOTALS
TAX DUE:	\$ 20,667.83	\$ 20,667.83	\$ 41,335.66
INTEREST DUE:			
TAX PAID:	(\$ 20,667.83)	(\$ 20,667.83)	(\$ 41,335.66)
PAID DATE:	10/03/2018	10/03/2018	
REMAINING AMOUNT:	\$0.00	\$0.00	\$0.00
TOTAL DUE:			\$0.00

Рисунок 1.1 — Пример таблицы из области налогообложения.

В литературе постановка задачи извлечения информации из таблиц (задача понимания таблиц) формулируется разным образом, и не существует единой типизации для такого рода задач. Прежде чем формально описать постановку задачи магистерской диссертации (глава 2), рассмотрим в данной главе существующие постановки смежных задач (раздел 1.2) и введем определения, связанные с задачей понимания таблиц (раздел 1.3). В разделе 1.4 будут описаны доступные размеченные коллекции таблиц, в разделе 1.5 произведем краткий обзор методов решения задач наиболее близких к задаче, рассматриваемой в данной диссертации, в разделе 1.6 будет описана значимость задачи исследования. В разделе 1.7 будет произведено обобщение и выделение ключевых моментов, которые имеют интерес для решения исследуемой задачи в магистерской диссертации.

1.2. Смежные задачи

В литературе можно найти множество задач, связанных с извлечением информации из таблиц (интерпретацией таблиц). При этом можно выделить три подхода к типизации задач: по Zang и Balog [10], по Jan C. и др. [8] и по Braunschweig [16]. Далее рассмотрим каждый из этих подходов.

1.2.1 Задачи извлечения информации из таблицы по Zang и Balog

Основными современными задачами извлечения информации из веб-таблиц выделяемые авторами статьи [10] являются:

1. извлечение таблиц (процесс идентификации и классификации таблиц);
2. интерпретация таблиц (раскрытие семантики, содержащейся в таблицах, с целью представления таблицы в структурированном виде, удобном для автоматической обработки);
3. поиск по таблицам (ответ на поисковый запрос ранжированным списком таблиц);
4. расширение базы знаний на основе данных из таблиц;
5. вопросно-ответные системы на основе информации из таблиц.

Задачи по извлечению таблиц и их интерпретации являются важными основополагающими задачами, на базе которых может быть построено решение высокоуровневых задач 3-5. В рамках магистерской диссертации интересны именно первые две задачи. Далее рассмотрим подробное описание задач 1 и 2, а в разделе 1.5 опишем существующие методы их решения.

Извлечение таблиц

Извлечение таблиц — это процесс идентификации и классификации таблиц в согласованный формат, в результате чего создается корпус таблиц. Сюда входят такие задачи, как *классификация реляционных таблиц*, *обнаружение заголовков* и *классификация типов таблиц*.

Идентификация таблиц из веб-страниц на основе разметки HTML обычно не вызывает затруднений. Однако таблицы в HTML также широко используются для целей форматирования и компоновки. Поэтому извлечение веб-таблиц включает в себя подзадачу очистки данных, то есть выявление и фильтрацию «плохих» таблиц (где «плохие» обычно означают неподлинные таблицы или не реляционные, см. подраздел 1.3.2). **Классификация реляционных таблиц** относится к задаче прогнозирования того, содержит ли веб-таблица реляционные данные (такие данные в таблице имеет схему, которая никогда не задана явно, но очевидна для наблюдателя и состоит из нескольких типизированных помеченных столбцов, рисунок 1.2).

Реляционные таблицы (рисунок 1.2) описывают набор сущностей вместе с их атрибутами и считаются высококачественными благодаря реляционным знаниям, которые они содержат. Однако, в отличие от таблиц в реляционных базах данных, эти отношения не являются явными в веб-таблицах, что представляет серьезную исследовательскую проблему. Большинство исследований фокусируются именно на таблицах такого типа, однако всего в Интернете 2-3% таблиц могут быть отнесены к реляционному типу.


№ ↕	Область (на белор.)	Площадь, км ² (2023) ^[138] ↕	Население, чел. (2023) ^[6] ↕	Административный центр ↕	Внутреннее деление ↕
1	Брестская (Брэсцкая)	32 776,60	↘ 1 315 405	 Брест	16 районов
2	Витебская (Віцебская)	40 062,33	↘ 1 091 948	 Витебск	21 район
3	Гомельская (Гомельская)	40 381,76	↘ 1 347 469	 Гомель	21 район
4	Гродненская (Гродзенская)	25 131,90	↘ 998 600	 Гродно	17 районов
5	Минская (Мінская)	39 834,99	↘ 1 462 021	 Минск	22 района
6	Могилёвская (Магілёўская)	29 086,83	↘ 989 703	 Могилёв	21 район
7	Минск (Мінск)	353,60	↘ 1 995 471	 Минск	9 районов

Рисунок 1.2 — Пример реляционной таблицы

Задача **обнаружения заголовка** часто решается как задача классификации: является ли строка/столбец заголовком. Обнаружение заголовков обычно решается вместе с двумя другими задачами и использует аналогичные признаки.






Задача **классификации** таблицы — это задача определения типа (или класса) таблицы в соответствии с предопределенной таксономии типов (различные таксономии рассмотрены в подразделе 1.3.2). Признаки, предназначенные для обнаружения заголовков и классификации реляционных таблиц, также могут применяться для *классификации таблиц*, как показано в предыдущих исследованиях Wang и Hu (2002) [3], Lehmborg и др. (2016) и Cafarella и др. (2008) [12].

Интерпретация таблиц

Интерпретация таблиц направлена на раскрытие семантики данных, содержащихся в таблице, с целью сделать табличные данные пригодными для

обработки машинами. См. рисунок 1.3 в качестве иллюстрации задач.

All-time (List of Grand Slam men's singles champions)

Rank	Player	Total	Years
1	 Roger Federer	20	2003–2018
2	 Rafael Nadal	17	2005–2018
3	 Pete Sampras	14	1990–2002
	 Novak Djokovic	14	2008–2018
5	 Roy Emerson	12	1961–1967

Annotations:

- (A) Person <http://dbpedia.org/ontology/Person> (arrow pointing to the Player column)
- (B) http://dbpedia.org/page/Rafael_Nadal (arrow pointing to Rafael Nadal's row)
- (C) <Peter_Sampras, careerYears, 1990-2002> (arrow pointing to Pete Sampras's row)

Рисунок 1.3 — Иллюстрация интерпретации таблицы: (А) идентификация типа столбца (В) связывание сущностей (С) извлечение отношений [10]

Определяют три основные задачи интерпретации таблиц:

- идентификация типа столбца, то есть нахождение и связывание столбца таблицы с типом сущности, которую он содержит (например, на рисунке 1.3 найден столбец и тип сущности, к которой он относится — “Человек”);
- связывание сущностей — распознавание упоминаний объектов в ячейках и связывание их с записями в справочной базе знаний;
- извлечение отношений — связывание пары столбцов в таблице с отношением, которое существует между их содержимым и/или извлечения информации об отношениях из табличных данных и представления их в новом формате (например триплетом “субъект-предикат-объект”).

1.2.2 Задачи извлечения информации из таблицы по Jan С и др.

Извлечение данных из таблиц по Jan С. и др. [8] — это процесс, который преобразует таблицу исходного документа в набор записей. Где запись — это структура данных, в которой отдельные данные в кортеже наделены семантикой посредством информации извлеченной из метаданных таблицы. Jan С. и др. упоминают статью [9], в которой авторы описывают, что процесс извлечения информации состоит из 5 этапов: локализация, сегментация, распознавание структуры, функциональный анализ, структурный анализ и интерпретация.

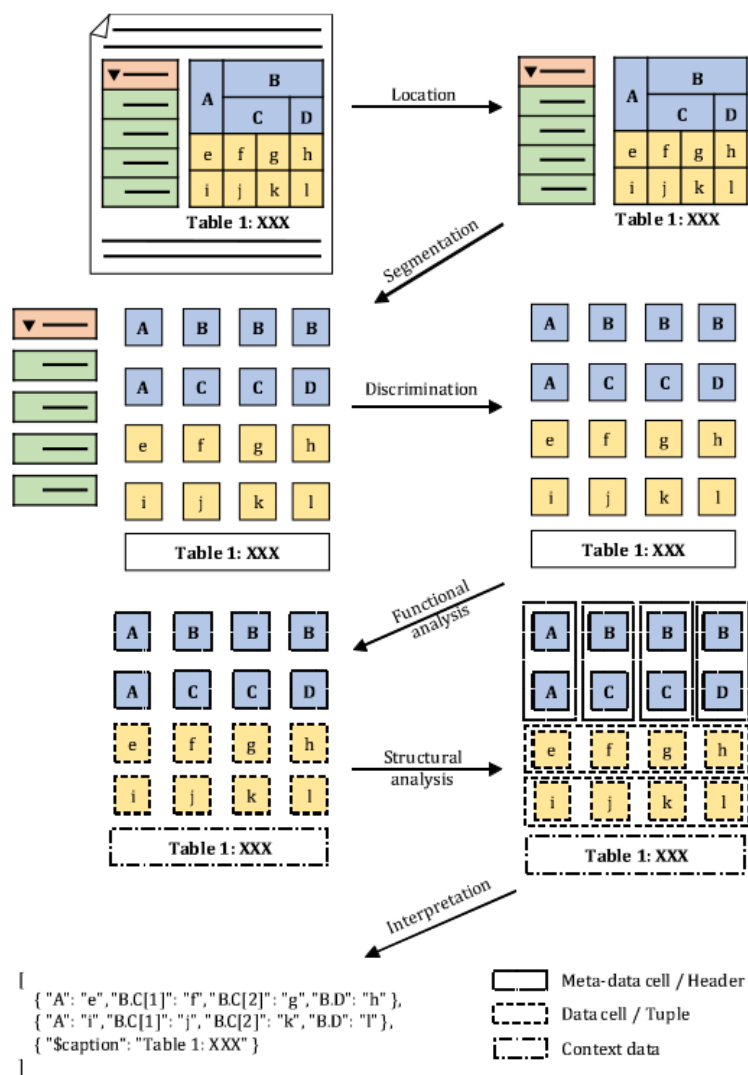


Рисунок 1.4 — Пример извлечения данных из таблицы [8]

На рисунке 1.4 показан пример процесса извлечения данных как последовательный конвейер.

Заметим, что авторы [8] в своем описании декомпозиции задачи извлечения информации рассматривают таблицы полученные из картинок или pdf-документов, поэтому локализация, сегментация являются задачами, которые решаются с таблицами, чья функциональная структура не была определена.

1.2.3 Задача понимания таблиц по Braunschweig

Автор статьи [16] рассуждает об автоматизированном понимании веб-таблиц как о развивающемся процессе, включающим в себя множество различных этапов обработки. На рисунке 1.5 показаны основные этапы.

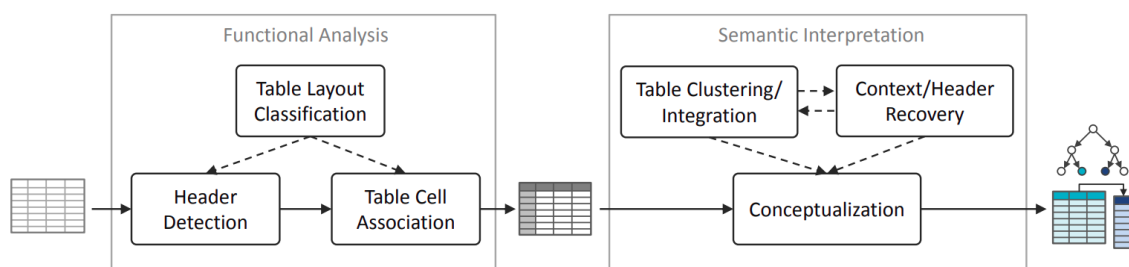


Рисунок 1.5 — Обзор процесса автоматизированного понимания таблиц [16]

Кроме упомянутых в предыдущих главах задачах определения типа таблицы и определения заголовка, автор также упоминает задачу **ассоциации ячеек таблицы**. Для таблиц с простыми двумерными схемами индексации, не имеющими вложенных или иерархических заголовков, связь между меткой и ячейками данных является простой. Каждая ячейка данных просто связана с метками в соответствующих заголовках строк и столбцов. Для более сложных схем индексации сначала необходимо разрешить структуру заголовка.

1.3 Таксономия таблиц

Таксономией является древообразная структура классификаций определенного набора объектов. Вверху этой структуры — объединяющая единая классификация — корневой таксон — которая относится ко всем объектам данной таксономии. Таксоны, находящиеся ниже корневого, являются более специфическими классификациями, которые относятся к поднаборам общего набора классифицируемых объектов.

Нетрудно видеть, что корневым таксоном в таксономии таблиц будет являться таблица (понятие таблицы приведено в подразделе 1.3.1). Следующим уровнем классификации разные исследователи выбирают разный принцип. Так, в работе [4] — это разделение на реляционные таблицы и таблицы компоновки, а в работе [6] — это разделение на одномерные, двумерные и многомерные таблицы.

1.3.1 Анатомия таблицы

Таблица **T** (рисунок 1.6) представляет собой сетку ячеек, расположенных в строках и столбцах. Таблицы используются в качестве шаблонов визуальной коммуникации, а также в качестве инструментов упорядочивания и организации данных.

Заголовок таблицы. Заголовок таблицы, T_H — список меток, определяющих содержание каждой строки/столбца таблицы. Заголовки обычно располагаются в первой строке/столбце таблицы.

Rank	Player	Total	Years
1	Roger Federer	20	2003–2018
2	Rafael Nadal	17	2005–2018
	Pete Sampras	14	1990–2002
	Novak Djokovic	14	2008–2018
5	Roy Emerson	12	1961–1967

Рисунок 1.6 — Состав таблицы [10]

Ячейка таблицы. Ячейка таблицы $T[i, j]$ указывается с индексом строки i и индексом столбца j . Ячейки таблицы содержат (возможно, пустые) значения и считаются атомарными единицами в таблице.

Строка таблицы. Строка таблицы $T[i, :]$ представляет собой список ячеек таблицы, лежащих горизонтально в строке i таблицы.

Столбец таблицы. Строка таблицы $T[:, j]$ представляет собой список ячеек таблицы, лежащих вертикально в столбце j таблицы.

Сущности таблицы. В таблицах часто упоминаются конкретные сущности, такие как лица, организации, местоположения. Сущности таблицы T_E — это множество, состоящее из всех сущностей, упомянутых в таблице.

1.3.2 Развитие таксономий таблиц

В литературе предложен ряд схем классификации таблиц.

Таблица 1.1 — Развитие таксономии типов таблиц

Источник	Количество типов	Типы таблиц
Wang и Hu, 2002 [3]	2	Подлинные (реляционные таблицы) и неподлинные (макетные таблицы)
Crestan и Pantel, 2011 [4]	11	1. Реляционные: таблица-форма, вертикальный список, горизонтальный список, таблица вида атрибут/значение, перечислительная таблица, календарь, матричная таблица, другие. 2. Таблицы компоновки: таблица форматирования, навигационная таблица.
Lautert и др., 2013 [5]	11 + 4	Добавлены: сжатые таблицы, вложенные таблицы, многозначные таблицы и разделенные таблицы (рисунки 1.7, 1.8)

Продолжение таблицы 1.1

Milosevic и др., 2016 [6]	3	Одномерные таблицы, двумерные таблицы и многомерные таблицы (таблицы супер строк и мульти таблицы)
Lehmberg и др. [7]	3	Реляционная, таблица-сущность, матричная таблица

В статье [3] 2002 года проводят различие между подлинными и неподлинными таблицами:

- **Подлинные** таблицы являются конечными таблицами, т. е. не содержат других таблиц, списков, форм, изображений или других элементов в ячейке; кроме того, они содержат несколько строк и столбцов;
- под **неподлинными** таблицами понимаются таблицы, которые не являются конечными и таблицы, которые используются для группировки содержимого для удобного просмотра.

Авторы статьи [4] разрабатывают детализированную таксономию, организованную в многоуровневую иерархию. Первый уровень иерархии — это деление на класс реляционных таблиц и класс таблиц компоновки:

1. **Реляционные таблицы** знаний содержат реляционные данные.
 - a. *Списки* относятся к таблицам, состоящим из ряда сущностей с одним атрибутом. С точки зрения направления макета они также классифицируются как вертикальные списки или горизонтальные списки.
 - b. *Таблицы атрибутов/значений* описывают определенный объект вместе с его атрибутами.
 - c. *Матричные таблицы* имеют в качестве заголовка первую строку и первый столбец. *Таблицы перечисления* перечисляют ряд объектов, которые имеют схожую природу.
 - d. *Таблицы форм* состоят из полей ввода, в которые пользователь может вводить или выбирать значения.
2. **Таблицы компоновки** не содержат никакой информации и используются только для целей компоновки.
 - a. *Навигационные таблицы* предназначены для навигации внутри или вне веб-сайта.
 - b. *Таблицы форматирования* используются для визуальной организации контента.

В статье [5] 2013 года уточняют схему классификации [4]. Реляционные таблицы дополнительно классифицируются в соответствии с таксономией второго порядка:

1. *Сжатые таблицы* (рисунок 1.7) содержат объединенные ячейки (т. е. ячейки с одним и тем же значением, объединенные вместе), чтобы избежать повторения значений.
2. *Вложенные таблицы* содержат таблицу в ячейке.
3. *Многозначные таблицы* относятся к таблицам, содержащим несколько значений в одной ячейке.
4. *Разделенные таблицы* (рисунок 1.8).

PLANT	COLOR	HEIGHT
SHRUBS		
Azalea	variable	shrub
Buddleia	blue, pink, white	shrub
CULTIVATED ANNUALS		
Alyssum	violet, white	4 inches

Year	Title
2010	<i>Death at a Funeral</i>
	<i>I Love You Too</i>
	<i>Pete Smalls Is Dead</i>
2011	<i>A Little Bit of Heaven</i>

(a)
(b)

Рисунок 1.7 — Примеры сжатых таблиц

Rank	City name	Pop.	Rank	City name	Pop.
1	São Paulo	11,316,149	6	Belo Horizonte	2,385,639
2	Rio de Janeiro	6,355,949	7	Manaus	1,832,423
3	Salvador	3,093,605	8	Curitiba	1,764,540
4	Brasília	2,609,997	9	Recife	1,536,934
5	Fortaleza	2,476,589	10	Porto Alegre	1,413,094

Рисунок 1.8 — Пример разделенной таблицы

В статье [7] Lehmberg и др. различают три основных типа таблиц:

1. **реляционные** таблицы содержат набор сущностей с их атрибутами; атрибуты могут находиться в строках или столбцах;
2. **таблицы сущностей** описывают определенную сущность;
3. **матричные таблицы** — таблицы с заголовками в виде строки и столбца.

Method/Features	AV	A	V
Our method (Window)	0.591	0.405	0.247
Our method (Naive)	0.197	0.250	0.102
Vector Space Model	0.0467	0.152	0.0128

Model	Test Accu
MV-RNN (Socher et al. 2013)	44.4
RNTN (Socher et al. 2013)	45.7
Bi-LSTM (Li et al. 2015)	49.8
Tree-LSTM (Tai, Socher, and Manning 2015)	51.0
CNN-non-static (Kim 2014)	48.0
CNN-Tensor (Lei, Barzilay, and Jaakkola 2015)	51.2
NCSL (Teng, Vo, and Zhang 2016)	51.1
LR-Bi-LSTM (Qian, Huang, and Zhu 2017)	50.6
Word Embedding with additive attention	47.47
Word Embedding with s2t self-attention	48.87
Multi-head with s2t self-attention	49.14
Bi-LSTM with s2t self-attention	49.95
DiSAN without directions	49.41
DiSAN	51.72

Рисунок 1.9 — Пример матричной и реляционной таблицы

1.4 Коллекции таблиц для задачи определения класса таблицы

В открытом доступе представлено ряд коллекций таблиц с размеченными классами таблиц и выделенными сущностями из таблиц. Однако, основная часть размеченных классов таких таблиц не покрывает все разнообразие таблиц. Отсутствуют такие классы, как сжатые, вложенные, разделенные таблицы и др.

Таблица 1.2 — Коллекции таблиц

Коллекция	Тип	Кол-во таблиц	Источник	Классы таблиц
WDC 2012 Web Table Corpus [7]	Web tables	147M	Web crawl (Common Crawl)	Реляционные(1.3%), нереляционные (классификация [3])
WDC 2015 Web Table Corpus	Web tables	233M	Web crawl (Common Crawl)	Реляционные, таблицы-сущностей, матричные таблицы (0.9%, 1.4%, 0.03%)
Dresden Web Tables Corpus	Web tables	174M	Web crawl (Common Crawl)	(Не в открытом доступе)
WebTables	Web tables	154M	Web crawl	Реляционные, таблицы-сущностей, матричные таблицы, “другие” (не в открытом доступе)
WikiTables	Wikipedia tables	1.6M	Wikipedia	Реляционные
DeepTable [15]	Web tables	5503	Научные статьи	Таблицы горизонтальной ориентацией, таблицы вертикальной ориентацией, матричные таблицы.

1.5 Обзор методов решения смежных задач

Методы решения задачи классификации реляционной таблицы

Первопроходцем-новатором в исследовании реляционных веб-таблиц является проект WebTables (Cafarella и др., 2008 [12]). Авторы статьи рассматривают реляционные таблицы как документы высокого качества и фильтруют их, обучая классификатор на основе правил. Классификатор

использует характеристики таблицы (признаки перечислены в таблицах 2.1, 2.3) в качестве признаков. Модель обучается на наборе аннотированных вручную таблиц: реляционных и нереляционных. В результате они создают высококачественный корпус, состоящий из 154 миллионов таблиц, отфильтрованных из 14,1 миллиарда HTML-таблиц.

Wang и Hu (2002) [3] определяют таблицу как подлинную, если она представляет собой листовую таблицу, в которой ни в одной из ячеек не существует подтаблицы. Они используют методы машинного обучения (деревья решений и SVM) для классификации реляционных таблиц, используя три основные группы признаков: признаки макета, признаки содержимого ячейки и признаки группы слов. Некоторые признаки макета и содержимого ячейки перечислены в таблицах 2.1, 2.3. Признаки групп слов вычисляются как статистики частотности слов.

Основываясь на [3], Eberius и др. (2015) [13] проводят классификацию реляционных таблиц, а также классификацию по типу макета (вертикальные списки, горизонтальные списки и матричные таблицы). Используются различные методы машинного обучения, включая деревья решений, случайные леса и SVM, с использованием комбинации глобальных и локальных признаков (некоторые признаки перечислены в таблицах 2.1-2.3 в главы 2). В результате Eberius и др. (2015) классифицируют миллионы таблиц и создают Dresden Web Table Corpus (DWTC, см. раздел 1.4).

Методы решения задачи классификации таблицы

Crestan и Pantel (2011) [4] собрали размеченную коллекцию из 8,2 миллиарда таблиц, используя глобальные признаки таблиц, признаки макета, содержимого и HTML (таблицы 2.1, 2.3). Глобальные признаки включают максимальное количество строк, столбцов и максимальную длину ячейки. Признаки макета включают среднюю длину ячеек, дисперсию длины и соотношение между строками и столбцами. Признаки содержимого включают признаки HTML (на основе тегов HTML) и лексические признаки (на основе содержимого ячейки). В качестве классификаторов в работе использовались Gradient Boosting и SVM. В корпус вошли следующие типы таблиц: таблица-форма, вертикальный список, горизонтальный список, таблица вида атрибут/значение, перечислительная таблица, календарь, матричная таблица. К сожалению, полученного корпуса нет в открытом доступе.

Как продолжение работы [4], Lautert и др. (2013) [5] рассмотрели расширение таксономии из работы [4]: многоуровневую таксономию. Первый уровень классификации аналогичен таксономии [4]. Второй уровень ориентирован на *сжатые таблицы, вложенные таблицы, многозначные*

таблицы и разделенные таблицы. Они использовали нейронную сеть с прямой связью для классификации таблиц.

Lehmberg и др. (2016) [7] создают коллекцию веб-таблиц из Common Crawl (WDC Web Table Corpus, см. 2.3.1). Для этого они отфильтровывают неподлинные таблицы (не самые внутренние таблицы, т. е. таблицы, содержащие в своих ячейках другие таблицы) и таблицы, содержащие менее двух столбцов или трех строк. Затем, используя объединение признаков из работ [3] и [4], отфильтрованные таблицы классифицируются как *реляционные, матричные, таблицы сущностей* или *таблицы компоновки*.

Методы глубокого обучения также использовались для классификации таблиц. Например, Nishida и др. (2017) [14] рассматривают таблицу как матрицу текста, аналогичную изображению. Они построили архитектуру нейронной сети под названием TabNet, с собственной таксономией типов таблиц (вертикальная и горизонтальная таблицы-сущности, горизонтальная и вертикальная реляционные таблицы, матричная таблица и “другие” таблицы). Структура сети состоит из кодировщика RNN, кодировщика CNN и классификатора. На первом этапе кодировщик RNN кодирует ячейки таблицы, чтобы создать трехмерный объем таблицы, аналогичный данным изображения. Затем трехмерный объем таблицы обрабатывается кодировщиками CNN для захвата семантики таблицы и глобальных признаков, который впоследствии используется классификатором для определения типов таблиц.

Авторы TabVec [17] используют подход обучения без учителя. Для этого они создают векторное представление ячеек основываясь на тексте из ячеек. Затем для набора ячеек количественно измеряется сходство ячеек в семантическом значении, вычисляя отклонение от среднего и отклонение от медианы для векторов ячеек. Далее векторное представление таблиц получается как конкатенация вектора отклонения от среднего и вектора отклонения от медианы. Затем используются методы кластеризации для определения класса таблицы (*таблица-сущность, таблица-список, матричная таблица, реляционная таблица, таблица без данных*).

Метод решения задачи ассоциации ячеек таблицы

Примером решения задачи ассоциации ячеек таблицы является работа [18] (2013). В работе используется трехступенчатый подход для выявления и разрешения иерархических атрибутов: поиск области данных/атрибутов, извлечение иерархии и построение кортежей (рисунок 1.10). На первом этапе идентифицируется область значений и область атрибутов. Далее для области атрибутов восстанавливаются иерархические связи между каждой ячейкой. Здесь авторы опираются на схожесть стиля форматирования и близость ячеек в геометрическом плане. Последний этап полностью зависит от результата

предыдущих двух и сработает правильно, если предыдущие шаги верно определили нахождение атрибутов и связи между ними. Алгоритм построения кортежей следующий. Для каждого значения v из области значений находятся его аннотирующие атрибуты по пути к корню в иерархиях атрибутов как для левой, так и для верхней областей атрибутов. Таким образом создается реляционный кортеж для каждого значения в области значений.

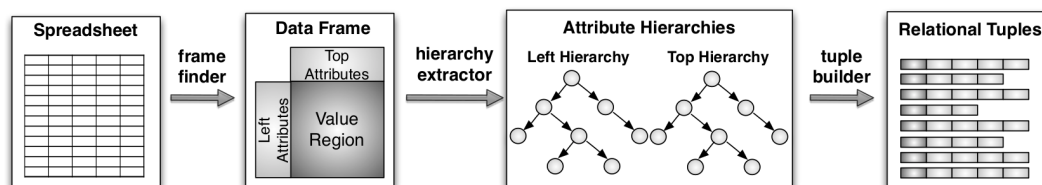


Рисунок 1.10 — Конвейер обработки электронной таблицы [18]

1.6 Значимость задачи исследования

Данные, представленные в разделе 1.5, демонстрируют наличие множества решений для таблиц ограниченного набора классов.

Согласно представленным данным в разделе 1.4, большинство коллекций таблиц, предоставленных в открытом доступе, состоят из таблиц реляционного типа. Заметим, что лишь 1-3% таблиц в Интернете относится к этому типу таблиц. Таким образом, исследователи измеряют качество своих алгоритмов в основном на закрытых коллекциях, что ставит под вопрос воспроизводимость их экспериментов.

Особенностью данной магистерской диссертации является использование таблиц из домена, который не представлен в известных исследованиях, а также применение методов машинного обучения для анализа таблиц сложной структуры.

В качестве источника данных взяты сайты с историей по налоговой активности. В них таблицы имеют вариативную структуру и ряд особенностей: большое количество пропусков (рисунок 1.1), различное расположение заголовочных строк и столбцов (рисунок 1.11), иерархические заголовки, сложные типы таблиц (рисунок 1.12).

Parcel	3106181050000	Default Date	2015-06-30	Tax Type	AS - Annual Secured
Bill	140877165*5	Extend Date	2014-12-03	Effective Date	2013-07-01
Corrected From		Corrected To		Eligibility	R - UNELIG REFUND DUE
Tax Rate Total	0.012142	Tax Rate Area	000012143	Tax Rate Year	2013

Рисунок 1.11 — Пример таблицы из области налогообложения.

Supplemental Property Tax			Note: Penalties only apply to late payments			
Parcel	Roll Year	Fiscal Year	Tax Rate Area	Value Date	Original Bill Date	
2017-002-147-030-7-01	2017	2017	05-001	06/05/2017	09/25/2018	
Owner Address		1 st Installment Delinquent After		2 nd Installment Delinquent After		
*Name private per CA AB2238		Dec 10, 2018		Apr 10, 2019		
201 FILBERT ST,STE #700		General Tax	Installment 1	Installment 2	Total	
SAN FRANCISCO CA 94133-0000			21,339.32	21,339.32	\$42,678.64	
Property Location		Penalty + Cost + Fee	0.00	0.00	\$0.00	
38 OAKLAWN DR		Total Amount	\$21,339.32	\$21,339.32	\$42,678.64	
DALY CITY		Paid Date	12/07/2018			
Base Values	Land	Improvement	Personal Property	Exemptions	Net Cash	Composite Rate
OLD	1,065,875	1,983,826	0	0	3,049,701	1.1714
NEW	1,147,904	5,545,186	0	0	6,693,090	Penalty Rate
SUPPLEMENTAL	82,029	3,561,360	0	0	3,643,389	10.0 %
PERIOD COVERED BY THIS STATEMENT: 07/01/2017 Through 06/30/2018						
Be aware that during peak periods, it may take up to 10 days to receive and process your payments.						

Рисунок 1.12 — Пример таблицы из области налогообложения

1.7 Выводы

В разделе 1.2 были изложены основные типы задач, имеющие отношение к предмету данной магистерской диссертации. В разделе 1.4 описаны существующие коллекции размеченных таблиц. В разделе 1.5 произведен краткий обзор методов решения задач классификации таблиц. Можно сделать следующие выводы.

1. В большинстве исследований фигурирует ограниченный список типов таблиц. Однако таблицы по налоговой активности имеют структуру, не лежащую на типичную таксономию из 5 типов: таблица-сущность, таблица-список, матричная таблица, реляционная таблица, таблица без данных.
2. В работе авторов прослеживается тенденция использования методов машинного обучения в качестве основных инструментов исследования.
3. Авторы [16] дают интересные идеи по преобразованию таблицы в набор кортежей: по типу ячеек проще вывести связь-ассоциацию между ячейками. Однако они учитывают только таблицы матричного типа и используют excel-таблицы в качестве источника.
4. Ряд задач решается над реляционными таблицами (идентификация столбца, связывание сущностей). Поэтому результаты работ, решающих такие задачи, сложно переносимы на задачу данного исследования.

5. Наиболее похожими задачами для текущего исследования являются:
 - a. определение класса таблицы, обнаружение заголовка (глава 1.2.1);
 - b. функциональный анализ, структурный анализ и интерпретация (глава 1.2.2);
 - c. задача ассоциации ячеек таблицы, и как сопутствующая задача — задача определения типа ячейки таблицы (глава 1.2.3).

ГЛАВА 2

РЕШЕНИЕ ЗАДАЧИ

В контексте магистерской диссертации извлечение информации из таблицы — это процесс, который преобразует таблицу в наборы записей. Где запись — это информация об объекте из таблицы в виде набора кортежей (рисунок 2.1).

Department	Country	Month		
		January	February	March
Sales	USA	A	B	C
	Belarus	D	E	F
Market	USA	G	H	I
	Other	J	K	L

→

(Department: Sales, Country: USA, Month: January, Value: A),
(Department: Sales, Country: USA, Month: February, Value: B),
(Department: Sales, Country: USA, Month: March, Value: C),
(Department: Sales, Country: Belarus, Month: January, Value: D),
(Department: Sales, Country: Belarus, Month: February, Value: E),
(Department: Sales, Country: Belarus, Month: March, Value: F),
...

Рисунок 2.1 — Процесс извлечения структурированной информации из таблицы
Задача извлечения информации из таблиц декомпозируется следующим образом:

- **этап 1** — выделение структуры таблицы;
- **этап 2** — извлечение семантически обогащенных записей по полученной структуре (далее *кортежей*), т.е. извлечение кортежей состоящих из пар атрибут-значение.

Существуют различные подходы для выполнения первого этапа. Эти подходы зависят от класса, к которому принадлежит таблица. В магистерской диссертации рассмотрена задача поиска структуры таблицы как задача определения класса таблицы (для таблиц простой структуры) и как задача определения класса ячеек в таблице (для таблиц сложной структуры).

Заметим, что при качественном выделении структуры таблицы, второй этап основывается на применении несложных правил. Например, если на первом этапе была решена задача определения класса таблицы, то по классу таблицы легко восстановить расположение заголовка. Зная расположение заголовка можно легко извлечь необходимые кортежи (рисунок 2.2).

На основе проведенного анализа было принято решение в магистерской диссертации основное внимание уделить задаче выделения структуры веб-таблицы.

Таким образом, **задача** исследования состоит в том чтобы применить современные методы машинного обучения по преобразованию полуструктурированных табличных данных в структурированные и оптимизировать существующие алгоритмы для решения задачи извлечения информации из веб-таблиц с данными по налогообложению.

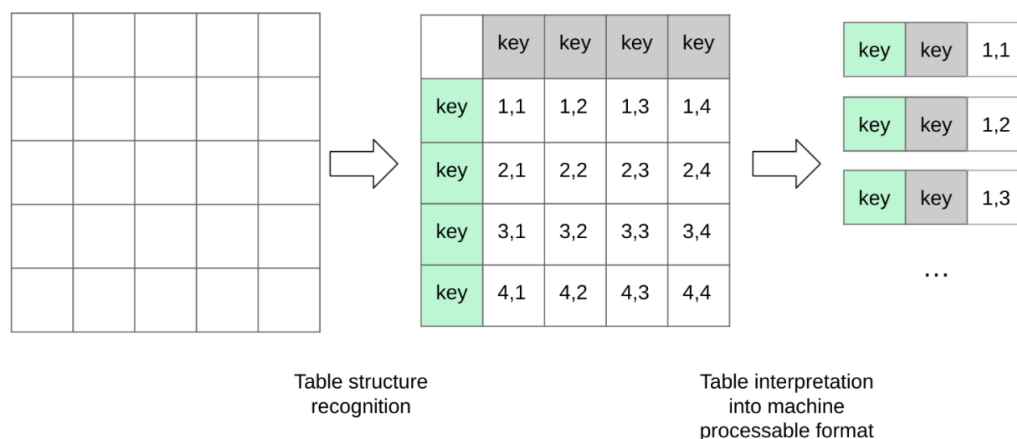


Рисунок 2.2 — Процесс извлечения информации из таблицы матричного типа

В следующем разделе 2.1 будут рассмотрены методы решения задачи выделения структуры таблицы, которые легли в основу практического исследования данной магистерской работы. Для задачи определения класса таблицы будут описаны эвристический подход, подход, использующий методы классического машинного обучения, и подход, использующий методы глубокого обучения. Для задачи определения класса ячейки будет описан метод, использующий глубокое обучение. В разделе 2.2 будет кратко рассмотрен детерминированный метод выделения кортежей из таблиц.

2.1 Выбранные методы решения задачи выделения структуры в таблицах

На основании раздела 1.4 можно сделать следующее обобщение, что таблицы могут иметь простую структуру и сложную. К таблицам **простой** структуры можно отнести таблицы, не имеющие иерархических заголовков и относящиеся к классам матричных таблиц, таблиц вертикальной и горизонтальной компоновки. К таблицам **сложной** структуры можно отнести сжатые, разделенные, вложенные таблицы и др.

Нетрудно видеть, что для выделения функциональных связей между ячейками таблиц простой структуры достаточно определить класс таблицы. Для определения функциональных связей у таблиц сложной структуры надо полностью восстановить роль каждой ячейки в таблице. Заметим, что для таблиц сложной структуры, имеющих иерархические заголовки, также необходимо восстановить связь между ячейками, в то время как в таблицах без иерархических заголовков связь можно восстановить однозначно по роли ячейки.

В магистерской диссертации были рассмотрены два подхода для решения задачи выделения структуры в таблицах:

1. определение класса таблицы (для таблиц простой структуры);
2. определение роли ячеек, также называемое определением типа или класса ячеек (для таблиц сложной структуры).

Далее в подразделах 2.1.1, 2.1.2, 2.1.3 будут описаны известные методы решения задачи 1; в подразделе 2.1.4 будут рассмотрены известные методы решения задачи 2 и подробно описан выбранный для экспериментов метод.

2.1.1 Эвристика в задаче определения класса таблицы

В работе [7] используется простая эвристика для обнаружения класса таблицы (класс горизонтально ориентированной, вертикально ориентированной таблицы и матричная таблица). Алгоритм основан на простой идее: в горизонтальных таблицах атрибуты представлены в столбцах, а в вертикальных таблицах атрибуты находятся в строках. Следовательно, для горизонтальных таблиц ячейки в одном столбце имеют одинаковую длину (количество символов в ячейке), потому что они принадлежат одному и тому же атрибуту. Напротив, для вертикальных таблиц справедливо обратное: ячейки в одной строке будут иметь одинаковую длину. Таким образом, они определяют ориентацию таблицы, вычисляя стандартное отклонение (*std*) длины ячейки для каждой строки и столбца. Если среднее *std* для столбцов значительно меньше среднего *std* для строк, таблица определяется как горизонтальная, если больше — как вертикальная, если же среднее *std* по столбца и по строкам равно — матричная.

2.1.2 Методы машинного обучения в задаче определения класса таблицы

Постановка задачи машинного обучения

Задача машинного обучения для классификации таблицы формулируется следующим образом. Имеется конечное множество таблиц, для которых известен их класс в соответствии с заданной таксономией. Такое множество называют множеством прецедентов или обучающей выборкой. На основе имеющихся данных требуется построить алгоритм, способный с высокой вероятностью выдавать правильный класс таблицы.

Признаки

В разделе 1.5 был произведен исторический разбор методов классического машинного обучения в задаче определения класса таблицы и смежных задачах. Стоит заметить, что признаки, используемые в задачах определения класса реляционной таблицы и обнаружения заголовка также могут быть использованы в рассматриваемой задаче. Ниже приведем компиляцию признаков, использующихся в литературе.

Рассматривая задачу определения класса таблицы и смежные задачи, можно выделить следующие типы признаков:

1. глобальные признаки макета (таблица 2.1);
2. локальные признаки макета (таблица 2.2);
3. признаки содержания ячейки (таблица 2.3).

Таблица 2.1 — Используемые глобальные признаки макета в задачах классификации типа таблицы (КТТ) и обнаружения заголовка (ОЗ)

Признак	Описание	Задача	Источник
Max cell length	Максимальное число символов в строке	КТТ	[4], [13]
#rows	Число строк	КТТ, ОЗ	[4], [13]
#cols	Число столбцов	КТТ, ОЗ	[4], [13]
%rows	Процент строк, которые содержат NULL	КТТ	[12]
μ	Средняя длина строки в ячейке	КТТ	[12]
δ	Стандартное отклонение длины строки ячейки	КТТ	[12]
$\frac{\mu}{\delta}$	Длина строки	КТТ	[12]
Avg cell length	Среднее количество символов в ячейке	КТТ	[12]
%length 1	Процент столбцов у которых: $ len(row_1) - \mu > 2\delta$	ОЗ	[12]
%length 2	Процент столбцов у которых: $\delta \leq len(row_1) - \mu \leq 2\delta$	ОЗ	[12]
%length 3	Процент столбцов у которых: $ len(row_1) - \mu < \delta$	ОЗ	[12]
Avg rows	Среднее количество ячеек в строках	КТТ	[3], [13]
Avg cols	Среднее количество ячеек в столбцах	КТТ	[3], [13]
Avg cell length	Среднее число символов в ячейке	КТТ	[3], [4], [13]

Таблица 2.2 — Используемые локальные признаки макета в задаче классификации реляционной таблицы (КРТ)

Признак	Описание	Задача	Источник
Std dev rows	Стандартное отклонение числа ячеек в строках	КРТ	[3], [13]

Продолжение таблицы 2.2

Std dev cols	Стандартное отклонение числа ячеек в колонках	КРТ	[3], [13]
Std dev cell length	Стандартное отклонение числа символов в ячейках	КРТ	[3], [13]
Local length avg	Средний размер ячеек в сегменте	КРТ	[3], [13]
Local length variance	Дисперсия размера ячеек в сегменте	КРТ	[3], [13]

Таблица 2.3 — Используемые признаки содержания ячейки для задач классификации типа таблицы (КТТ), классификации реляционной таблицы (КРТ) и обнаружения заголовка (ОЗ)

Признак	Описание	Задача	Источник
%body non-string	Процент нестроковых данных в теле таблице	ОЗ	[12]
%header non-string	Процент нестроковых данных в первой строке	ОЗ	[12]
%header punctuation	Процент столбцов со знаками препинания в первой строке	ОЗ	[12]
Local span ratio	Соотношение ячеек с тегом 	КТТ, КРТ	[4], [13]
Local ratio header	Соотношение ячеек с тегом <th>	КТТ, КРТ	[4], [13]
Local ratio anchor	Соотношение ячеек с тегом <a>	КТТ, КРТ	[4], [13]
Ratio img	Соотношение ячеек, содержащих изображения	КТТ, КРТ	[3], [4], [13]
Ratio form	Соотношение ячеек, содержащих формы	КТТ, КРТ	[3], [4]
Ratio alphabetic	Соотношение ячеек, содержащих буквенные символы	КТТ, КРТ	[3], [4]
Ratio digit	Соотношение ячеек, содержащих цифры	КТТ, КРТ	[3], [4]
Ratio empty	Соотношение пустых ячеек	КТТ, КРТ	[3], [4]

Методы машинного обучения

Далее в таблице 2.4 будет представлено перечисление методов машинного обучения, применяемых в задаче определения класса таблицы.

Таблица 2.4 — Классы таблиц и методы классического машинного обучения

Классы таблиц	Методы	Источник
Подлинные, неподлинные	Решающие деревья, SVM	[3]
Реляционные, нереляционные	Классификатор, основанный на правилах	[12]
Таблица форматирования, навигационная таблица, таблица-форма, вертикальный/горизонтальный список, таблица вида атрибут/значение, перечислительная таблица, календарь, матричная таблица, другие	Градиентный бустинг	[4]
Вертикальные списки, горизонтальные списки, матричные таблицы	Решающие деревья, SVM	[13]

2.1.3 Методы глубокого обучения в задаче определения класса таблицы

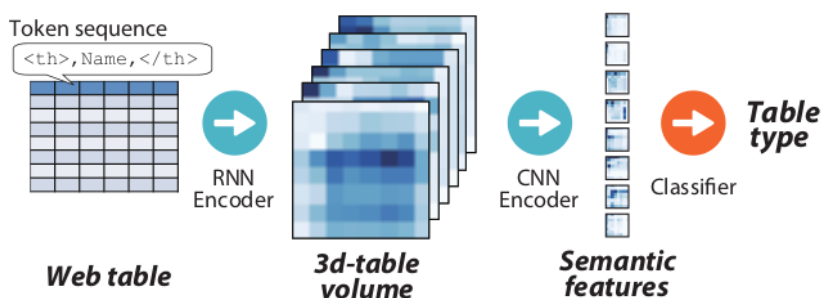


Рисунок 2.3 — Концепция TabNet. RNN сначала кодирует последовательность токенов в каждой ячейке, а затем CNN извлекает семантические признаки для классификации типа таблицы [14]

Нейронная сеть TabNet (рисунок 2.3), описанная в статье [14] Nishida и др., долгое время являлась золотым стандартом для задачи определения класса таблицы, где структура таблиц простая, а таксономия классов включает в себя следующие классы:

- вертикальная таблица-сущность;
- горизонтальная таблица-сущность;
- горизонтальная реляционная таблица;

- вертикальная реляционная таблица;
- матричная таблица;
- “другие” таблицы.

Однако авторы статьи [14] не предоставили доступ к их обучающей коллекции данных, что затрудняет сравнение результатов экспериментов. Поэтому было принято решение рассмотреть архитектуру DeepTable авторов Nabibi и др. [15]. Nabibi и др. показывают в своей статье, что архитектура DeepTable является более эффективной в сравнении с архитектурой TabNet на их собственной коллекции таблиц, извлеченных из научных статей.

Концепция сети DeepTable включает следующие операции (рисунок 2.4):

1. Получение **векторного представления ячейки** — $u_{c_i,j}$. Эмбединг ячейки получается путем извлечения токенов w_i из содержимого текста ячейки, получения векторного представления e_i для каждого токена с помощью предварительно обученного автоэнкодера и последующего применения двунаправленной сети LSTM (Bi-LSTM) для набора векторов e_i и получения на выходе сети Bi-LSTM двух векторов — $h_{T_{c_i,j}}, h'_{T_{c_i,j}}$. Затем многослойный перцептрон (MLP) используется для отображения выходных данных слоя Bi-LSTM в новое векторное пространство, чтобы извлечь как нелинейную связь между двумя представлениями LSTM, так и связь между ячейками, так как MLP является общим для всех ячеек.
2. Получение **векторного представления таблицы “по столбцам”** и **“по строкам”**. Оба представления получаются путем применения аналогичных операций. Ниже приведем подробное описание получения векторного представления “по столбцам”. Сначала операция субдискретизации (pooling) применяется к набору векторных представлений ячеек $(u_{c_1,j}, \dots, u_{c_N,j})$ в столбце j , так для столбца j получается агрегированное представление l_j . Затем полученные векторы проходят через общую сеть MLP на выходе из которой получается обогащенный вектор u_{l_j} столбца j . Этот этап извлекает взаимосвязь между столбцами в представлении по столбцам, поскольку слой MLP используется всеми столбцами совместно. Таким образом получаются обогащенные наборы векторов $(u_{l_1}, \dots, u_{l_M})$, конкатенирует которые получается векторное представление таблицы по столбцам.
3. Для получения метки класса таблицы векторное представление таблицы “по строкам” и “по столбцам” конкатенируются и подается на вход трехслойному линейному классификатору с функцией активацией softmax.

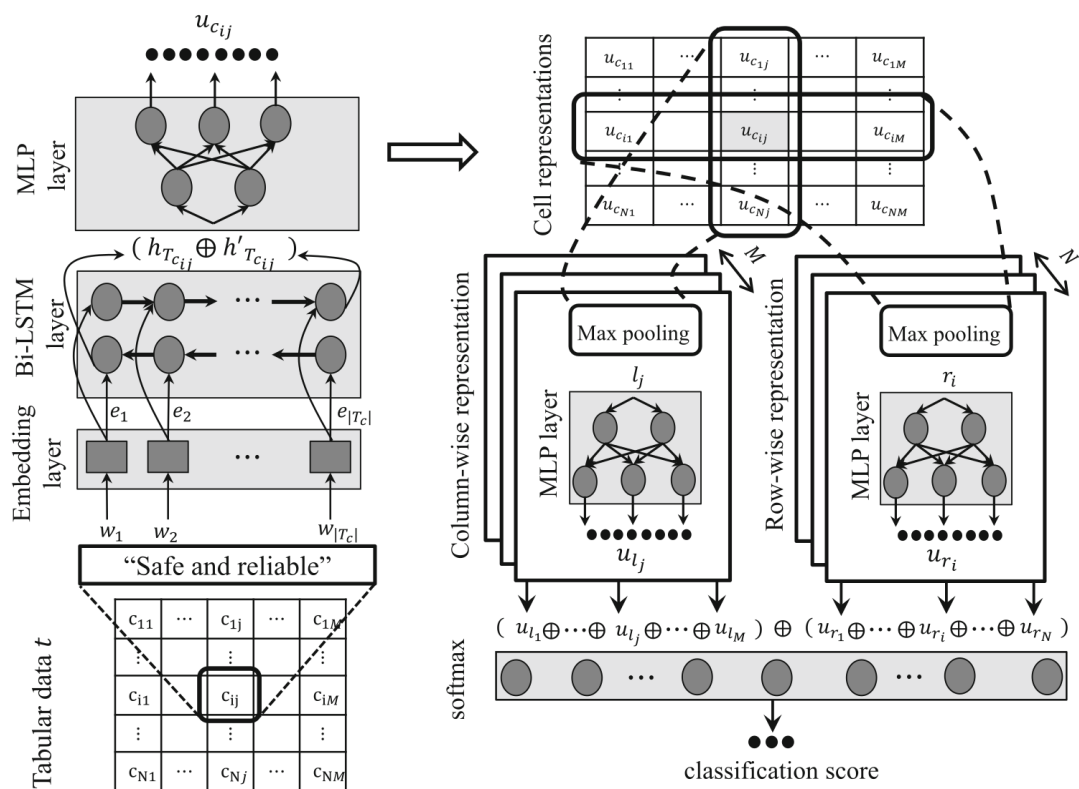


Рисунок 2.4 — Концепция DeepTable. Пример содержимого ячейки «Safe and reliable» токенизирован и представлен в виде эмбедингов e_i [15]

2.1.4 Методы машинного обучения в задаче определения роли ячеек в таблице

В статье 2020 года [20] и 2021 года [21] были описаны архитектуры нейронных сетей TUTA и TabularNet, соответственно, решающие задачу определения класса ячейки путем создания векторного представления ячеек. Авторы оценивают свои методы в сравнении с методами, представленными ранее в литературе, на собственных коллекциях и утверждают, что их подходы показывают более точные результаты. В обеих работах обучающие коллекции состоят из таблиц матричного типа. Архитектура TUTA использует механизмы трансформеров, что усложняет использование данного метода, в сочетании с тем фактом, что обучающая выборка из статьи [20] содержит таблицы со структурой, не похожей на структуру таблиц, рассматриваемых в данной магистерской диссертации. Статья [21] использует рекуррентные нейронные сети, что облегчает обучение на коллекциях с различной природой. Тем не менее, авторы не предоставили доступ к исходному коду своей работы. Поэтому за основу экспериментов была взята работа 2019 года [19]. Ниже будет рассмотрен метод, описанный в статье [19], который использовался в качестве основы экспериментов в данной магистерской работе.

Классификация ячеек таблицы с использованием предварительно обученных векторных представлений ячеек

В статье [19] метод классификации ячеек состоит из двух шагов. Авторы сначала создают модель для генерации векторных представлений ячеек в табличных документах. На втором шаге они разрабатывают и обучают классификатор на основе RNN, который использует эти векторные представления для определения классов ячеек.

Модель векторного представления ячейки состоит из двух частей: первая часть выражает глобальную семантическую информацию, используя семантическое содержание ячеек, в то время как вторая часть представляет локальную информацию из скрытых паттернов стиливых признаков каждой ячейки. Метод классификации принимает во внимание зависимости между ячейками в документе, анализируя последовательности ячеек в каждой строке и столбце. Обзор системы показан на рисунке 2.5.

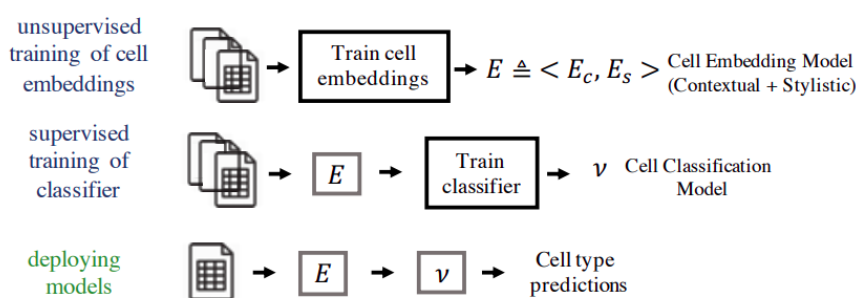


Рисунок 2.5 — Обзор системы классификации ячеек [19]

В статье [19] авторы стремятся создать систему без учителя, которая научится извлекать **векторные представления ячеек** из неразмеченных табличных документов (таблиц). Более формально, задан документ D , представленный в виде табличной матрицы с N строками и M столбцами, $D = \{C_{i,j}; 1 \leq i \leq N, 1 \leq j \leq M\}$, определим коллекцию ячеек как $(C_{i,j})$. Авторы хотят научиться векторизовать ячейку $C_{i,j}$ и ее контекст с помощью эмбединг-оператора (E) в k -мерный вектор, $V_{i,j} \in \mathbb{R}^k$. В данной статье оператор E состоит из двух частей. Первая часть представляет глобальную семантическую информацию ячейки таблицы, используя ее текстовое содержимое и контекст (E_c). Вторая часть представляет локальную информацию из скрытых паттернов стиливых характеристик каждой ячейки (E_s). Авторы определяют эмбединг-оператор ячейки как конкатенацию операторов семантического и стилистического эмбединг операторов, т.е.

$E = \langle E_c, E_s \rangle$. Далее опишем подробнее механизм работы синтаксического и стилистического эмбединг-операторов.

Семантическое векторное представление ячейки (E_c)

Сам по себе текст в ячейке не содержит достаточно информации о ячейке и ее роли в таблице. Чтобы извлечь осмысленное представление ячейки, авторы считают что ее контекст также должен учитываться. Можно заметить что контекст ячейки бывает двух видов: локальный (клетки окружающие рассматриваемую клетку) и глобальный (зависимость между ячейками на далеком расстоянии). Авторы замечают, что для определения глобального контекста нужна информация о классе таблицы (т.е. ее типе компоновки), что не возможно учесть в методе обучения без учителя, поэтому они ограничились рассмотрением только локального контекста ячейки. Формально, локальный контекст целевой ячейки $C_{i,j}$ в табличном документе D определяется как $X_{C_{i,j}} = C_{i-2,j}, C_{i-1,j}, C_{i+1,j}, C_{i+2,j}, C_{i,j-2}, C_{i,j-1}, C_{i,j+1}, C_{i,j+2}$.

Рисунок 2.6 показывает обзор модуля по выделению семантического векторного представления ячейки. Модуль состоит из двух частей. Верхняя сеть (E_c^{ctx}) предсказывает значение целевой ячейки, учитывая значения ячеек контекста, а нижняя сеть (E_c^t) предсказывает значение ячеек контекста, учитывая значение целевой ячейки.

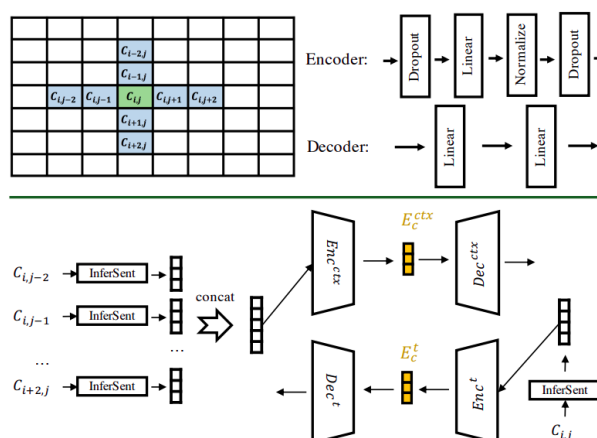


Рисунок 2.6 — Обзор системы выделения семантического векторного представления ячейки [19]

Для кодирования текстовой информации ячейки авторы использовали предварительно обученную модель InferSent, которая позволяет получить векторное представление предложений. Для кодирования слов использовались предварительно обученные векторные представления из модели GloVe. Стоит заметить, что слова, на которых модель GloVe не была обучена, обрабатываются

модулем кодирования предложений как неизвестные, поэтому многие числовые значения будут неизвестны для модели InferSent. Авторы отмечают, что кодирование числовых значений не дало прироста в точности классификатора, поэтому они не рассматривали данный подход.

Для формального определения предложенной системы кодирования авторы вводят определение InferSent модуля как функции I , принимающей текстовое значение ячейки, и возвращающей d -мерное векторное представление: $I : \mathbb{S} \rightarrow \mathbb{R}^d$, где \mathbb{S} — множество всех предложений. Также вводятся понятия модулей кодирования и декодирования как $Enc^{ctx} : \mathbb{R}^{8d} \rightarrow \mathbb{R}^{d'}$, $Enc^t : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$, $Dec^{ctx} : \mathbb{R}^{d'} \rightarrow \mathbb{R}^d$, $Dec^t : \mathbb{R}^{d'} \rightarrow \mathbb{R}^d$. Здесь d' — размерность скрытого выхода модуля кодирования, которая предполагается одинаковой для E_c^{ctx} и E_c^t . Входом для Enc^{ctx} является конкатенация векторов контекста ячейки размерности $8d$.

Нейронные сети E_{ctx} и E_t обучаются отдельно. При обучении минимизируется функцию потерь, которая представляет собой среднюю квадратическую ошибку выходных данных сети и целевого вектора. Для E_{ctx} целевым вектором будет являться векторное представление целевой ячейки, а для сети E_t — конкатенация векторного представление ячеек контекста. Функция потерь для сети E_{ctx} , E_t имеет вид:

$$l^{ctx}(\phi) = \sum_i \left| I(C_i) - Dec_{\phi_1}^{ctx} \left(Enc_{\phi_2}^{ctx} \left(I(X_{C_i}) \right) \right) \right|^2 \quad (2.1)$$

$$l^t(\phi) = \sum_i \sum_{C_j \in X_{C_i}} \left| I(C_j) - Dec_{\phi_3}^t \left(Enc_{\phi_4}^t \left(I(C_i) \right) \right) \right|^2 \quad (2.2)$$

где $\phi = \langle \phi_1, \phi_2, \phi_3, \phi_4 \rangle$ — параметры сети, i — номер ячейки в тренировочном корпусе. X_{C_i} — это набор ячеек локального контекста для C_i , а $I(X_{C_i})$ — это конкатенация выходных векторов модуля InferSent для значений ячеек локального контекста. Таким образом, обучение нейронных сетей сводится к поиску параметров, минимизирующих функцию потерь, т.е. $argmin_{\phi} l^{ctx}(\phi) + l^t(\phi)$.

Итоговое семантическое векторное представление ячейки получается путем конкатенации выхода модуля Enc^{ctx} сети E_c^{ctx} и модуля Enc^t сети E_c^t : $E_c(C_{i,j}, X_{C_{i,j}}) = \langle Enc^{ctx}(X_{C_{i,j}}), Enc^t(C_{i,j}) \rangle$.

Стилистическое векторное представление ячеек таблицы

Веб-таблицы содержат обширную информацию о форматировании в ячейках, т.е. различные признаки текста (например, наличие заглавных букв, наличие чисел, количество ведущих пробелов) и признаки оформления ячеек (например, размер шрифта, цвет шрифта, цвет фона, тип границ). Чтобы использовать эту информацию, сначала создается целочисленное представление всех категориальных признаков и получается целочисленный вектор, представляющий характеристики ячейки. Затем эти целочисленные векторы преобразуются в непрерывные числовые векторы (**стилистическое векторное представление**) с помощью механизма автокодировщика. Авторы статьи объясняют значимость этого шага тем, что нельзя использовать целочисленные векторы вместе с семантическим векторным представлением в качестве входа для рекуррентного классификатора. Заметим, что такое использование возможно, но может ухудшать точность классификации.

Сеть автокодировщика (рисунок 2.7), состоящая из энкодера(кодировщика) и декодера (де-кодировщика), восстанавливает входной целочисленный вектор (декодер) и генерирует непрерывные векторные представления (энкодер), которые используются как **стилистическое векторное представление** (E_s) в системе классификации [19]. Во время обучения авторы статьи используют среднеквадратичную ошибку между выходом декодера и истинным целочисленным вектором в качестве функции потерь.

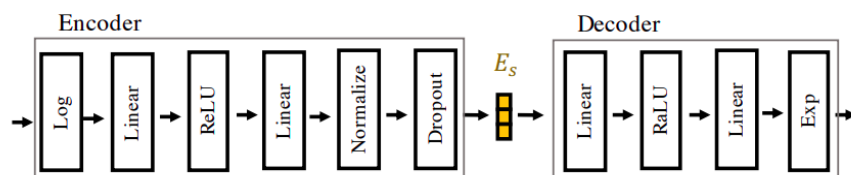


Рисунок 2.7 — Сеть автокодировщика для выделения стилистического векторного представления ячейки [19]

Классификация ячеек таблицы

В статье [19] впервые применяются рекуррентные нейронные сети (RNN) для определения класса ячеек, в то время как ранее RNN использовались только для определения класса таблицы.

Для определения класса ячеек Majid Ghasemi-Gol и др. [19] используют блоки LSTM для захвата зависимостей между ячейками в таблицах. Блок LSTM наблюдает последовательность входных векторов (x_1, x_2, \dots, x_n) и генерирует скрытый выход — вектор h_i для каждого вектора последовательности, где x_i —

это векторное представление i -ой ячейки в коллекции. Блок LSTM поддерживает внутреннее состояние, и для каждого вектора в последовательности входов скрытый выход LSTM является функцией его состояния, входного вектора и его предыдущего выхода. LSTM поддерживает информацию о произвольных точках ранее в последовательности и способен захватывать долгосрочные зависимости. Это особенно полезно для захвата информации об удаленном контексте ячеек, который не учитывается в рамках схемы векторного представления ячеек. Например, за верхним заголовком может следовать большая последовательность ячеек с данными в колонке, поэтому может быть полезно для классификатора помнить о наличии верхнего заголовка при классификации последующих ячеек с данными.

Таблицы предполагают наличие зависимостей между ячейками как в строках, так и в столбцах. Для учета обеих зависимостей авторы объединяют две сети LSTM, одна из которых наблюдает за последовательностью ячеек в каждой строке (назовем ее $LSTM_{row}$), а другая — за последовательностью ячеек в каждом столбце (назовем ее $LSTM_{col}$). Эта архитектура дает возможность LSTM-блокам учитывать ячейки слева и сверху от целевой ячейки при генерации выходного вектора h_i для нее. Напомним, что в качестве входных данных для LSTM-сетей используется векторное представление введенное в предыдущем разделе.

На рисунке 2.8 показан обзор схемы классификации ячеек. Для заданной таблицы с N строками и M столбцами сначала создается вектор x_i векторное представление ячейки, как это было объяснено в предыдущем пункте. Получается тензор размером $N \times M$. Затем производится добавление мнимых строк и столбцов, чтобы выделить границы документа. В результате получается тензор (T_D) размером $(N + 2) \times (M + 2) \times k$, где k — размерность векторного представления ячейки. Для документа будет $N + 2$ последовательностей строк и $M + 2$ последовательностей столбцов.

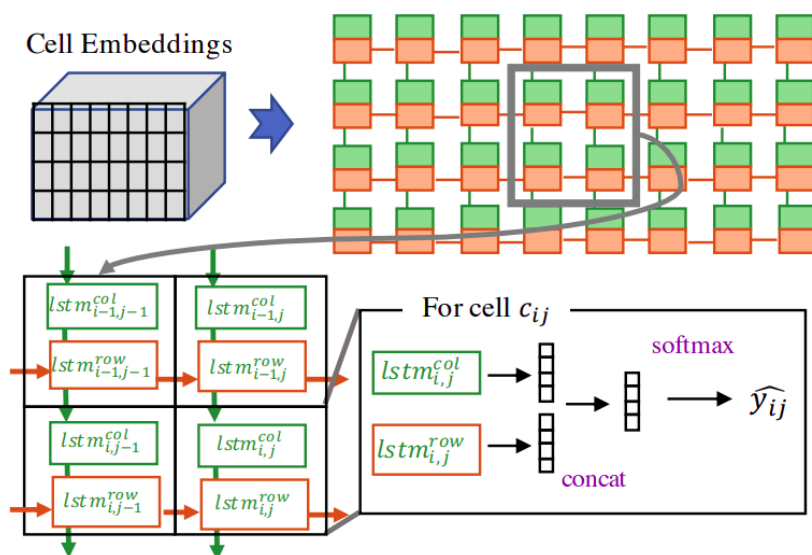


Рисунок 2.8 — Метод классификации основанный на рекуррентных нейронных сетях [19]

Чтобы проиллюстрировать принцип работы классификационного фреймворка, рассмотрим ячейку в строке i и столбце j в тензоре, созданном в результате процесса дополнения мнимых строк/столбцов (это соответствует ячейке в строке $i-1$ и столбце $j-1$ в исходной таблице), и назовем ее целевой ячейкой. Для классификации целевой ячейки сеть $LSTM_{row}$ наблюдает строку i , а сеть $LSTM_{col}$ наблюдает столбец j в T_D . Кроме того, скрытый выход j из $LSTM_{row}$ (h_j^r), и выход i из $LSTM_{col}$ (h_i^c) соответствуют целевой ячейке. Далее объединяются эти два вектора и используются линейный слой для уменьшения размерности от $2k$ до количества классов ячеек K . Затем используется слой softmax для расчета вероятностей разных классов целевой ячейки. Запишем более формально:

$$\hat{y}_{i,j}^{\phi_r, \phi_c} = (h_j^{r, \phi_r}, h_i^{c, \phi_c})\theta^T + b \quad (2.3)$$

$$\hat{p}_{i,j}(k; \phi_r, \phi_c, \theta) = \frac{e^{\hat{y}_{i,j}^{k, \phi_r, \phi_c}}}{\sum_{k=1}^K e^{\hat{y}_{i,j}^{k, \phi_r, \phi_c}}} \quad (2.4)$$

где $\hat{y}_{i,j}^{\phi_r, \phi_c}$ — это выход линейного слоя размером K ; $\hat{p}_{i,j}(k; \phi_r, \phi_c, \theta)$ — это k -ый выход softmax слоя; ϕ_r, ϕ_c, θ - параметры $LSTM_{row}$, $LSTM_{col}$ и линейного слоя соответственно.

Авторы используют взвешенную отрицательную логарифмическую функцию правдоподобия в качестве функции потерь для обучения классификационной сети. Функция потерь может быть формально записана как:

$$l(\phi_r, \phi_c, \theta) = - \sum_{d_i} \sum_{i,j} \sum_{k=1}^K w_k y_{i,j,d_i}^k \log(\hat{p}_{i,j}^{k;\phi_r,\phi_c,\theta}) \quad (2.5)$$

где d_i — индекс документа в обучающей коллекции, i — индекс строки, j — индекс столбца, k — индекс класса ячейки, w_k — вес метки k , $\hat{p}_{i,j}^{k;\phi_r,\phi_c,\theta}$ задается уравнением (2.4), y_{i,j,d_i}^k — вектор длиной K , в котором единица находится только в позиции, соответствующей истинному значению целевой ячейки, а во всех остальных позициях — ноль. Значение w_k устанавливается обратно пропорциональным количеству ячеек с типом класса k в обучающей коллекции

$$(n_k^{train}): w_k = 1 - \frac{n_k^{train}}{\sum_{k'=1}^K n_{k'}^{train}}.$$

Таким образом обучение классификационного фреймворка сводится к поиску параметров минимизирующих функцию потерь (2.5), т.е. $argmin_{\phi_1,\phi_2,\theta} l(\phi_1, \phi_2, \theta)$

При тестировании на новом документе класс ячейки для каждой ячейки в таблице вычисляется с помощью уравнения (2.4), и выбирается класс ячейки с максимальной вероятностью, т.е. $argmax_k \hat{p}_{i,j}(k)$.

2.2 Алгоритм выделения кортежей из таблицы


Как описано в разделе 2.1, выделение кортежей из таблицы **простого** типа является тривиальной задачей, но в зависимости от класса будет иметь свою особенность:

1. если это таблица горизонтальной или вертикальной ориентации, то извлекаемый кортеж будет иметь вид: список пар атрибут-значение;
2. если класс таблицы матричный — (атрибут 1, атрибут 2, значение);

Для выделения кортежей из таблиц **сложного** типа рассмотрим следующие случаи: таблица с иерархическими заголовками, разделенные и сжатые таблицы без иерархических заголовков .

Чтобы извлечь кортежи из таблицы с *иерархическими* заголовками, надо преобразовать ее в таблицу простого типа горизонтальной ориентации (рисунок 2.9). Не трудно видеть, что определив роль ячейки, т.е. входит ли она в состав вертикального или горизонтального заголовка, данное преобразование легко выполнимо, т.к. связь между ячейками иерархического заголовка определяется соседством.

Department	Country	Month		
		January	February	March
Sales	USA	A	B	C
	Belarus	D	E	F
Market	USA	G	H	I
	Other	J	K	L




Department	Country	Month	Value
Sales	USA	January	A
Sales	USA	February	B
Sales	USA	March	C
Sales	Belarus	January	D
Sales	Belarus	February	E
Sales	Belarus	March	F
Market	USA	January	G
Market	USA	February	H
Market	USA	March	I
Market	Other	January	J
Market	Other	February	K
Market	Other	March	L

Рисунок 2.9 — Процесс упрощения структуры таблицы с иерархическими вертикальными и горизонтальными заголовками

Чтобы извлечь кортежи из *разделенной* таблицы, надо разделить ее на простые таблицы. Нетрудно видеть, что если в таблице встречается горизонтальные/вертикальные заголовки несколько раз и не граничат друг с другом, то таблица является разделенной. Нахождение простых таблиц, составляющих такую таблицу, не составляет труда (рисунок 2.10).

Country	Population	Area	Country	Population	Area
USA	A	B	Italy	G	H
Belarus	C	D	Finland	I	J
Greece	E	F	Togo	K	L




Country	Population	Area
USA	A	B
Belarus	C	D
Greece	E	F

Country	Population	Area
Italy	G	H
Finland	I	J
Togo	K	L

Рисунок 2.10 — Процесс упрощения структуры таблицы с иерархическими вертикальными и горизонтальными заголовками

Процесс извлечения кортежей из *сжатой* таблицы показан на рисунке 2.11.

Plant	Color	Height
Shrubs		
Azalea	variable	shrub
Buddleia	blue, pink, white	shrub
Cultivated annuals		
Alyssum	violet, white	4 inches



```
[
  (Shrubs, Plant: Azalea, Color: variable, Height: shrub),
  (Shrubs, Plant: Buddleia, Color: "blue, pink, white", Height: shrub),
  (Cultivated annuals, Plant: Alyssum, Color: "violet, white", ...)
]
```

Рисунок 2.11 — Процесс извлечения кортежей из сжатой таблицы

ГЛАВА 3

ПРИМЕНЕНИЕ МЕТОДОВ ОПРЕДЕЛЕНИЯ КЛАССА ТАБЛИЦЫ

3.1 Состав данных для обучения и сравнения моделей

Таксономия.

В магистерской диссертации рассмотрена таксономия, основанная на классификации из статьи [7]:

- таблицы горизонтальной ориентацией;
- таблицы с вертикальной ориентацией;
- матричные таблицы.

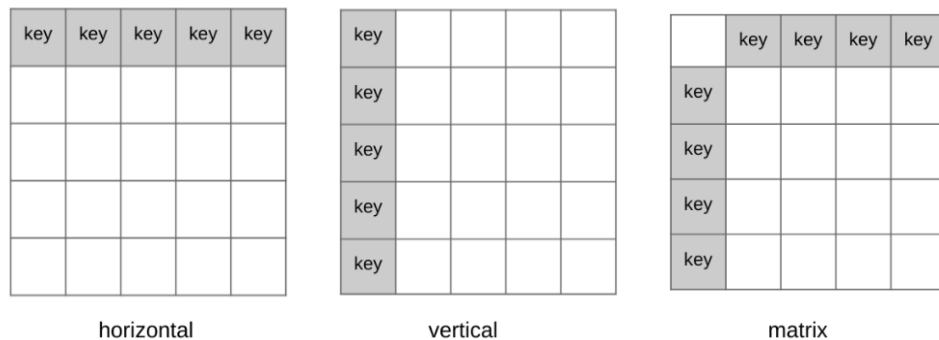


Рисунок 3.1 — Таксономия таблиц магистерской диссертации (слева на право визуальное представление таблицы горизонтальной, вертикальной ориентации и матричной таблицы)

Обучающая выборка.

Рассматривается две коллекции таблиц.

1. Коллекция DeepTable собранная авторами [15] (основана на данных из научных статей). Разметка данного корпуса была получена эвристически на основании признаков HTML разметки.
2. Коллекция таблиц TaxTable полученная из веб-страниц по налогообложению. Коллекция размечена вручную.

Таблица 3.1 — Количество размеченных таблиц по классам

Класс	Коллекция DeepTable	Коллекция TaxTable
Горизонтальная ориентация	1835	34
Вертикальная ориентация	1834	13
Матричная	1834	26

3.2 Подготовка данных

Для обучения и сравнения моделей классификации необходимо произвести разбиение данных на два множества:

- обучающую выборку, которая будет использоваться для подбора параметров моделей;
- тестовую выборку, на котором будут производиться замеры и сравниваться результаты.

В рамках магистерской диссертации для экспериментов с градиентным бустингом было взято 75% от каждого набора данных для обучения и 25% для валидации. Для экспериментов с нейронными сетями корпус DeepTable был использован как базовый набор данных, на которых производится обучение, так как корпус по налогообложению недостаточно большой для обучения нейронной сети.

3.3 Применение эвристики

Была применена простая эвристика из подраздела 2.1.1, основанная на усредненном стандартном отклонении длины строки в ячейке по столбцам и строкам. Поскольку для применения эвристики нет необходимости в использовании обучающей выборки, точность классификатора основанного на эвристики проверялась на целых коллекциях.

Таблица 3.2 — Точность классификатора основанного на эвристике

Коллекция	Случайный выбор	Эвристика
DeepTable	34.1%	47%
TaxTable	32.9%	38.4%

Не трудно видеть, что эвристика дает прирост точности по сравнению с случайным выбором. Таким образом, можно сделать вывод, что количество символов в ячейки и вариативность этого признака по строкам и столбцам — это важный признак при определении класса таблицы.

3.4 Применение градиентного бустинга

3.4.1 Признаковое описание таблицы

В магистерской диссертации используются два вида признаков: признаки, основанные на длине строки в ячейке, и признаки, основанные на содержании ячейки.

Обоснованием важности этих признаков может служить гипотеза о том, что человек может определить класс таблицы визуально без учета семантики таблицы. Так признаки длины строки в ячейке отображают визуальную информацию, т.е. зная количество символов в ячейках таблицы мы для некоторых видов таблиц можем однозначно определить ее класс.

В таблице 3.3 перечислим признаки используемые в магистерской диссертации.

Таблица 3.3 — Набор стилистических признаков классификатора

Признак	Описание	Подробности
% is num	Процент ячеек состоящих только из числа	Применен к 1,2 строке/столбцу
% contain num	Процент ячеек содержащих цифры	Применен к 1,2 строке/столбцу
% is null	Процент пустых ячеек	Применен к 1,2 строке/столбцу
% is unique	Процент уникальных ячеек	Применяется к 1 и 2 строке/столбцу
Std len	Стандартное отклонение длины строки в ячейке	Применен ко всем строкам/столбцам. Затем считается статистика второго порядка: std, mean, median
Mean len	Среднее длины строки в ячейке	Применен ко всем строкам/столбцам. Затем считается статистика второго порядка: std, median
Median len	Медиана длины строки в ячейке	Применен ко всем строкам/столбцам. Затем считается статистика второго порядка: std, mean, median

3.4.2 Обучение модели и полученные результаты

Для экспериментов была выбрана библиотека `sklearn`, использовался класс `GradientBoostingClassifier`, позволяющий обучать модели мультиклассовой классификации.

Было обучено два классификатора использующих метод градиентного бустинга и признаки из таблицы 3.3: один на тренировочной выборке коллекции DeepTable, второй на тренировочной выборке коллекции TaxTable.

Итоговая точность классификации на тестовой выборке коллекции DeepTable оказалась равной 78.7%, на тестовой выборке коллекции TaxTable – 73.68%.

3.5 Применение нейронных сетей

В качестве базовой архитектуры нейронной сети была выбрана архитектура DeepTable [15], подробное описание которой было приведено в подразделе 2.1.3.

Поскольку эксперимент с градиентным бустингом с признаками из таблицы 3.3 на коллекции DeepTable показал точность лучше на 5.29%, чем в оригинальной статье [15], было принято решение сделать модификацию архитектуры DeepTable — DeepTable с использованием признаков длины строки в ячейке и содержимого ячейки (далее **DeepTable на стилистических признаках**). В предложенной модификации этап получения векторного представления ячейки $u_{c_{i,j}}$, как отображения семантической информации ячейки, был заменен на представление ячейки вектором, содержащим стилистические признаки. В таблице 3.4 перечислим используемые признаки.

Таблица 3.4 — Стилистические признаки ячейки $c_{i,j}$

Признак	Описание	Признак	Описание
len	Длина ячейки	len mean by col/row	Среднее значение длины содержимого ячейки в <i>i</i> -ой строке/ <i>j</i> -ом столбце
is_empty	Является ли ячейка пустой	len median by col/row	Медиана длины содержимого ячейки в <i>i</i> -ой строке/ <i>j</i> -ом столбце
is_num	Является ли ячейка числом	count_not_null by col/row	Количество не пустых ячеек в <i>i</i> -ой строке/ <i>j</i> -ом столбце
contains_num	Содержит ли ячейка число	count_num by col/row	Количество чисел в <i>i</i> -ой строке/ <i>j</i> -ом столбце

Продолжение таблицы 3.4

len std by col/row	Стандартное отклонение длины содержимого ячейки в i-ой строке/ j-ом столбце	% unique by col/row	Процент уникальных ячеек в i-ой строке/ j-ом столбце
--------------------	---	---------------------	--

Таким образом были обучены нейронная сеть DeepTable и модификация сети DeepTable с стилистическими признаками. При обучении использовался корпус DeepTable. Код реализации сети DeepTable и параметры для обучения взяты из оригинальной статьи [15]. Далее в таблицах 3.5-3.7 приведем результаты полученные на коллекциях DeepTable и TaxTable.

Таблица 3.5 — Результаты нейронной сети DeepTable. В качестве тестирующего набора использовалась коллекция DeepTable

	Точность (precision)	Полнота	F1-мера	Кол-во
	0.7331	0.7948	0.7627	463
	0.7577	0.7446	0.7511	462
	0.7106	0.6631	0.6860	463
точность (accuracy)			0.7341	1388
macro avg	0.7338	0.7342	0.7333	1388
weighted avg	0.7338	0.7341	0.7333	1388

Таблица 3.6 — Результаты нейронной DeepTable на стилистических признаках. В качестве тестирующего набора использовалась коллекция DeepTable

	Точность (precision)	Полнота	F1-мера	Кол-во
	0.7148	0.9201	0.8045	463
	0.7751	0.8355	0.8042	462
	0.9762	0.6199	0.7583	463
точность (accuracy)			0.7918	1388
macro avg	0.8220	0.7918	0.7890	1388

Продолжение таблицы 3.6

weighted avg	0.8221	0.7918	0.7890	1388
--------------	--------	--------	--------	------

Таблица 3.7 — Результаты нейронной DeepTable на стилистических признаках. В качестве тестирующего набора использовалась коллекция TaxTable

	Точность (precision)	Полнота	F1-мера	Кол-во
	0.9286	0.3824	0.5417	34
	0	0	0	13
	0.4138	0.9231	0.5714	26
точность (accuracy)			0.5068	73
macro avg	0.4475	0.4351	0.3710	73
weighted avg	0.5799	0.5068	0.4558	73

3.6 Выводы

Заметим, что поскольку тестовые выборки обеих коллекций являются сбалансированными, в качестве метрики качества можно использовать точность (accuracy).

Основные результаты тестирования классификаторов на тестовой выборке коллекции TaxTable (таблица 3.8):

- более простой метод градиентного бустинга со стилистическими признаками показал лучший результат;
- нейронная сеть DeepTable со стилистическими признаками, обученная на коллекции DeepTable без дообучения на коллекции TaxTable, не показывает удовлетворительного качества.

Основные результаты тестирования классификаторов на тестовой выборке коллекции DeepTable из статьи [15] (таблица 3.8):

- метод градиентного бустинга со стилистическими признаками показал себя лучше, чем архитектура DeepTable статьи [15];
- модификация сети DeepTable — DeepTable со стилистическими признаками показала лучший результат.

Таблица 3.8 — Сравнение точности (ассурасу) классификаторов

Коллекция	Случайный выбор	Эвристика	Градиентный бустинг	Нейронная сеть DeepTable [15]	Нейронная сеть DeepTable со стилистическим и признаками
DeepTable	34.1%	47%	78.7%	73.41%	79.18%
TaxTable	32.9%	38.4%	73.68%	-	50.68%

Таким образом, получены следующие результаты.

1. Построен классификатор на основе градиентного бустинга, который дает удовлетворительное качество на тестовой выборке коллекции TaxTable, сравнимое с качеством, достигнутым в статье [15].
2. Предложена модификация метода DeepTable [15] для определения класса таблицы, которая на корпусе таблиц из научных статей показывает результат лучше, чем в оригинальной статье, а также использует более простые признаки, что ускоряет время обучения нейронной сети и время тестирования.

На основании проведенных экспериментов можно сделать следующие **ВЫВОДЫ**.

1. Эксперименты на коллекции DeepTable показали, что стилистические признаки вносят больший вклад в понимание класса таблицы в сравнении с семантическими признаками. Однако заметим, что в литературе последних лет наблюдается тенденция рассмотрения семантического представления без учета визуально-стилистического.
2. Нейросетевой подход, основанный на стилистических признаках, без дообучения не позволяет получить удовлетворительное качество на таблицах из домена, не представленного в тренировочной выборке. Таким образом можно заключить, что домен, из которого была получена таблица, определяет не только семантическую составляющую, но и визуально-стилистическую.

ГЛАВА 4

ПРИМЕНЕНИЕ МЕТОДОВ ОПРЕДЕЛЕНИЯ КЛАССА ЯЧЕЕК

4.1 Состав данных для обучения и сравнения моделей

Коллекции таблиц представленные в открытом доступе

Существующие коллекции таблиц, представленные в открытом доступе, DeEx, SAUS и CIUS для задачи определения класса ячеек в таблице собраны из разных разных источников (финансы, бизнес, сельское хозяйство, здравоохранение и т. д.), поэтому содержат таблицы с различной структурой и семантикой. В таблице 4.1 представлены данные о размерах используемых коллекций. Примеры таблиц представлены в приложении А.

Таблица 4.1 — Статистическое описание используемых коллекций таблиц

	SUAS	DeEX	CIUS	TaxDataset
Число аннотированных таблиц	223	444	248	137
Число аннотированных ячеек	192k	711k	216k	3704
Среднее число строк	52.5	220.2	68.4	8.8
Среднее число столбцов	17.7	12.7	12.7	3.7
% иерархических таблиц	93.7%	43.7%	72.1%	22.8%

Таксономия типов клеток в данных коллекциях классифицируют ячейки по общим типам: метаданные (metadata), примечания (notes), данные (data), заголовок (header), левый атрибут (attributes) и агрегированные значения(derived) (см. таблицу 4.2). Приведем описание шести классов ячеек:

1. данные — значения лежат в области данных;
2. агрегированные значения — агрегированные значения указывают, что некоторые значения вычисляются из других значений («итого» — это специальная агрегация, указывающая на операцию суммирования);
3. метаданные — как правило объясняют содержание документа;
4. примечания — предоставляют дополнительную информацию о документе или части документа;
5. заголовок (или верхний атрибут);
6. левый атрибут (или заголовок строк).

Подготовленная коллекция таблиц TaxDataset

Для упрощения процесса разметки ячеек был подготовлено приложение, написанное на языке python с использованием фреймворка ruqt5.

С различных сайтов было скачано и отфильтровано 68 уникальных таблиц. Общее количество сайтов — 12. С помощью аугментаций из них было получено еще 69 таблиц. Итого авторская коллекция насчитывает 137 аннотированных таблиц. Классы ячеек взятые из статьи [19]: метаданные (metadata), данные (data), заголовок (header), левый атрибут (attributes).

Таблица 4.2 — Распределение ячеек по разным классам

Коллекция таблиц	Классы ячеек					
	Атрибуты	Данные	Заголовок	Метаданные	Агрегированные значения	Примечания
DeEX+ SAUS	15091	630068	13221	9766	7172	2624
TaxDataset	453	2240	381	7	0	0

Заметим, что обе коллекции таблиц сталкиваются с проблемой несбалансированного распределения классов. Это необходимо учесть при обучении классификатора, используя функцию потерь, учитывающую распределения весов.

4.2 Подготовка данных

Для сравнения результатов эксперимента с результатами работы [19] были отобраны только пара коллекций DeEX и SAUS для первого этапа обучения моделей. Для обучения и сравнения моделей классификации коллекция была разбита на три множества в примерном процентном соотношении 80%/10%/10%:

- обучающую выборку, которая будет использоваться для подбора параметров моделей;
- валидационный набор, на котором будут производиться замеры и подбираться гиперпараметры модели;
- тестовый набор, на котором будет определена итоговая оценка качества модели.

Разбиение объединения коллекций DeEX, SAUS:

- обучающая выборка — 535;
- валидационная выборка — 66;
- тестовая выборка — 66.

Разбиение коллекции TaxDataset:

- обучающая выборка — 108;
- валидационная выборка — 15;
- тестовая выборка — 14.

Отметим, что таблицы коллекции TaxDataset были разделены так, чтобы в тестовой/валидационной выборке не было таблиц извлеченных с сайтов представленных в обучающей выборке и наоборот.

4.3 Применение нейронных сетей

В качестве классификатора был использован ансамбль моделей, описанный в подразделе 2.1.4 с небольшой модификацией, улучшающей качество. В работе [19] авторы исключали числа в ячейках при получении семантического векторного представления, однако в случае таблиц, относящихся к налогообложению, такой подход может привести к значительной потере информации. Поэтому было принято решение при создании семантического векторного представления E_c , токены состоящие из чисел не выбрасывать, а кодировать методом Фурье описанным в статье [22].

Было применено два подхода к обучению ансамбля моделей. Первый подход, известный как трансферное обучение (transfer learning), включает в себя обучение ансамбля на большой коллекции DeEX+SEUS, а затем дообучение ансамбля на более маленьком наборе данных TaxDataset в течение нескольких итераций. Второй подход включает в себя обучение ансамбля моделей сразу на тренировочной выборке коллекции TaxDataset.

Поскольку рассматриваемые коллекции не являются сбалансированными по классам, в качестве метрики оценки качества была выбрана метрика f1-macro-score.

4.3.1 Transfer learning: обучение на коллекциях DeEX, SEUS

Выбранные параметры

Для первого этапа трансферного обучения параметры были выбраны как в оригинальной статье[19]:

- размерность эмбединга предложения (предопределен модулем InferSent):
 $l = 4096$;
- размерность контекстного эмбединга ячейки: $d = 200$;
- размерность стилистического эмбединга ячейки: $d'' = 30$;

- количество эпох с преждевременной остановкой обучения(при не уменьшающейся функции потерь более 5 эпох, рассчитанной на валидационной выборке): 100;
- размер пакета: 200 ячеек;
- оптимизатор Adam со скоростью обучения 0,0005;
- вероятность выключения нейрона для слоев DropOut: $p = 0,1$;
- скорость обучения (learning rate): 0.0005.

Для второго этапа трансферного обучения параметры те же, кроме скорости обучения (0.0002) и числа эпох: 10 для обучения энкодеров и 2 для обучения классификатора. Количество эпох выбиралось исходя из f1-macro-score на валидационной выборке.

Результаты

На тестовой выборке f1-macro = **0.61**. Матрица ошибок (классы по порядку: attributes, data, header, metadata):

```
[[ 68  32   4   0]
 [  3 135   7   0]
 [  0   2  29   0]
 [  1   1   0   0]]
```

Таблица 4.3 — Результаты метода transfer learning на тестовой выборке коллекции TaxDataset

	Точность (precision)	Полнота	F1-мера	Кол-во
attributes	0.94	0.65	0.77	104
data	0.79	0.93	0.86	145
header	0.72	0.94	0.82	31
metadata	0	0	0	2
точность (accuracy)	0.82			282
macro avg	0.62	0.63	0.61	282
weighted avg	0.84	0.82	0.82	282

4.3.2 Обучение на коллекции таблиц TaxDataset

Выбранные параметры

В результате поиска гипер-параметров улучшающих f1-score, были выбраны:

- размерность эмбединга предложения (предопределен модулем InferSent): $l = 4096$;
- размерность контекстного эмбединга ячейки: $d = 200$;
- размерность стилистического эмбединга ячейки: $d'' = 30$;
- количество эпох: 58;
- размер пакета: 200 ячеек;
- оптимизатор Adam со скоростью обучения 0,0005;
- скорость обучения (learning rate): 0,0005;
- вероятность выключения нейрона для слоев DropOut: $p = 0,1$.

Результаты

На тестовой выборке $f1\text{-macro} = 0.86$. Матрица ошибок (классы по порядку: attributes, data, header, metadata):

```
[[ 97  1  6  0]
 [ 2 129 13  1]
 [ 0  1 30  0]
 [ 0  0  0  2]]
```

Таблица 4.3 — Результаты обучения нейронной сети с модификациями из статьи [19] на обучающей выборке TaxDataset на тестовой выборке коллекции TaxDataset

	Точность (precision)	Полнота	F1-мера	Кол-во
attributes	0.98	0.93	0.96	104
data	0.98	0.89	0.93	145
header	0.61	0.97	0.75	31
metadata	0.67	1	0.8	2
точность (accuracy)	0.91			282
macro avg	0.81	0.95	0.86	282
weighted avg	0.94	0.91	0.92	282

4.4 Выводы

В рамках данной главы был рассмотрен метод машинного обучения для определения класса ячеек таблиц сложной структуры из коллекции TaxDataset. В качестве основного метода была использована нейросетевая архитектура из статьи [19], которая была обучена двумя подходами: transfer learning с использованием коллекций DeEx и SAUS и обучение на коллекции TaxDataset.

Результаты первого подхода, оцененные метрикой f1-macro score, оказались ниже, чем результаты второго подхода, составив 0.61 и 0.82 соответственно.

Для адаптирования нейросетевой архитектуры под налоговый домен была добавлена модификация: представление чисел в виде эмбедингов, основанное на преобразовании Фурье. Использование данной модификации привело к улучшению качества по сравнению с методом, не включающим кодирование чисел: метрика f1-macro-score составила 0.86.

В приложении В представлены примеры работы классификатора на таблицах из коллекции TaxDataset. Анализ ошибок показал, что классификатор хорошо справляется с таблицами разделенного, матричного типа. Однако классификатор допускает ошибки на составных таблицах, состоящих из более чем двух таблиц.

Для дальнейшего улучшения классификатора могут быть применены следующие подходы:

- изучение влияния конфигурации контекста при формировании семантического представления ячейки;
- добавление признаков, использованных в главе 3 магистерской диссертации, в процессе создания стилистического представления.

ЗАКЛЮЧЕНИЕ

В рамках магистерской диссертации было произведено исследование существующих подходов решения задачи извлечения информации из таблиц по налогообложению. Был разработан метод, состоящий из двух этапов: выделение структуры таблицы и выделение семантически обогащенных кортежей. Для решения первого этапа были предложены методы определения класса таблицы (для таблиц простой структуры) и определения класса ячеек (для таблиц сложной структуры). Для задачи выделения структуры таблицы были рассмотрены и улучшены существующие методы из статей [15], [19]. Поскольку качество решения второго этапа напрямую зависит от первого, был кратко описан подход для решения проблемы выделения кортежей из таблицы.

В первой главе были рассмотрены смежные задачи и подходы к их решению.

Во второй главе формулируется постановка задачи, исследуемой в магистерской диссертации и подробно описываются методы машинного обучения для решения подзадач и их практическая реализация.

В третьей и четвертой главах приведены числовые результаты полученные при тестировании разработанных методов и описаны модификации рассматриваемых методов. Также описан состав коллекций TaxTable и TaxDataset подготовленных в рамках магистерской диссертации.

В результате можно сделать вывод, что, несмотря на свою актуальность и востребованность, задача извлечения информации из табличных данных не решается в необходимом объеме существующими методами из открытых источников и обладает большим потенциалом для исследования и оптимизации.

ПРИЛОЖЕНИЕ А

Building Design	Normal	Residence Class	Lot	CURRENT STATUS AND ...		
Grade	Good	Condition	Good	FIRST INSTALLMENT	SECOND INSTALLMENT	
Year Built	2000	# Stories	3	TAX DUE:	\$ 1,671.67	\$ 1,671.67
Total Area (sq. ft.)	1100	Main Floor Area	555	INTEREST DUE:		
# Rooms	10	# Bedrooms	3	TAX PAID:	(\$ 1,671.67)	
# 1/2 Baths	2	# Full Baths	1	PAID DATE:	09/28/2018	
Unfinished %		Unfinished Area	0	REMAINING AMOUNT:	\$0.00	\$ 1,671.67
Heat Type	Hotwater	Air Conditioning	No	TOTAL DUE:	\$ 1,671.67	

TAXING ENTITY	RATE	TOTAL TAXES DUE	AMOUNT PAID	BALANCE	Real Property Information	Account	Tax Year	Status
INSTALLMENT 1		Last Paid Date: 12-14-2017				06-45-24-C1-00427.0240	2018	PAID
JC FIRE #1	0.012440	\$119.46	\$119.46	\$0.00	Original Account	06-45-24-C1-00427.0240	Book/Page	2113/4744
JO CO LIBRARY	0.003921	\$37.65	\$37.65	\$0.00	Physical Address	807 SE 34TH ST CAPE CORAL ...	Owner	HOGAN JOHN M + JOANNE
JO CO PARK	0.003112	\$29.89	\$29.89	\$0.00	Legal Description	CAPE CORAL UNIT 15 BLK ...	Outstanding Balance as of ...	\$0.00
JOHNSON CO	0.019318	\$185.50	\$185.50	\$0.00	Mailing Address	807 SE 34TH STCAPE CORA...		
STATE OF KS	0.001500	\$14.40	\$14.40	\$0.00				
INSTALLMENT 2		Last Paid Date: 5-8-2018						
GARDNER CITY	0.020540	\$197.23	\$197.23	\$0.00				
JC FIRE #1	0.012440	\$119.46	\$119.46	\$0.00				
JO CO LIBRARY	0.003921	\$37.65	\$37.65	\$0.00				

TAXING ENTITY	TOTAL TAXES DUE	AMOUNT PAID	BALANCE
Installment 1			
231 BOND	\$546.80	\$546.80	\$0.00
231 SCH GEN	\$289.80	\$289.80	\$0.00
231 UNIFIED	\$501.00	\$501.00	\$0.00
COMM COLLEGE	\$147.51	\$147.51	\$0.00
GARDNER CITY	\$522.84	\$522.84	\$0.00
JO CO LIBRARY	\$52.87	\$52.87	\$0.00
JO CO PARK	\$39.35	\$39.35	\$0.00
JOHNSON CO	\$297.47	\$297.47	\$0.00
STATE OF KS	\$25.19	\$25.19	\$0.00
Installment 2			
TOTALS	\$2,422.83	\$2,422.83	\$0.00
Last Paid Date: 12-27-2012			

Real Property Information	Account	Tax Year	Status	Original Account	Book/Page
	06-45-24-C1-00427.0240	2018	PAID	06-45-24-C1-00427.0240	2113/4744
Physical Address	807 SE 34TH ST CAPE CORAL ...				
Owner	HOGAN JOHN M + JOANNE				
Legal Description	CAPE CORAL UNIT 15 BLK ...				
Outstanding Balance as of ...	\$0.00				
Mailing Address	807 SE 34TH STCAPE CORA...				

Рисунок 1 — Примеры таблиц из коллекции TaxDataset. Цвета отображают класс ячеек

ПРИЛОЖЕНИЕ Б

A												
1	2	3	4	5	6	7	8	9	10	11	12	13
Security/Ticker	Trade Date	Settlement Date	Instrument	Cost	Position	Strike Price	Units	Notional Value	Date	Price	Terminations	Units
MD												
PUBLICS												
3	08/03/00	08/03/03	Swap	\$	Long	\$ 1.18	78,000	\$ 91,937				
4	08/03/00	08/03/03	Swap	\$	Long	\$ 53.00	1,276,483	\$ 67,648,299	01/16/01	\$ 25.27	255,276	
5	08/03/00	08/03/03	Swap	\$	Long	\$ 162.50	1,093,426	\$ 177,681,725	01/11/01	\$ 30.44	1,000	
6	08/03/00	08/03/03	Swap	\$	Long	\$ 4.20	1,562,250	\$ 655,532				
7	08/03/00	08/03/03	Swap	\$	Long	\$ 1.339,286	116,115,000					
8	08/03/00	08/03/03	Swap	\$	Long	\$ 5.88	59,891	\$ 351,860				
9	08/03/00	08/03/03	Swap	\$	Long	\$ 1.68	735,000	\$ 1,237,703	11/09/00	\$ 1.94	735,000	
10	08/03/00	08/03/03	Swap	\$	Long	\$ 14.04	14,000	\$ 142,287				
11	08/03/00	08/03/03	Swap	\$	Long	\$ 4.07	127,500	\$ 518,400	12/14/00	\$ 7.00	127,500	
12	08/03/00	08/03/03	Swap	\$	Long	\$ 7.63	804,243	\$ 6,132,353	12/08/00	\$ 6.72	804,243	
13	08/03/00	08/03/03	Swap	\$	Long	\$ 7.63	804,243	\$ 6,132,353	12/08/00	\$ 6.72	804,243	
LA												
Full-time Law Enforcement Employees												
MD												
by Population Group												
Percent Male and Female, 2007												
TA												
Percentages												
Percent law enforcement employees												
Male												
Female												
Total												
Percent officers												
Male												
Female												
Total												
Percent civilians												
Male												
Female												
Total												
Number of												
officers												
civilians												
estimated												
2007												
LA												
TOTAL CITIES												
GROUP I (250,000 and over)												
GROUP II (50,000 to 249,999)												
GROUP III (50,000 to 49,999)												
GROUP IV (25,000 to 49,999)												
GROUP V (10,000 to 24,999)												
GROUP VI (under 10,000)												
NONMETROPOLITAN COUNTIES												
METROPOLITAN COUNTIES												
SUBURBAN AREA												
URBAN AREA												
Metropolitan area includes all metropolitan agencies answered with a Principal City. The agencies answered with Principal City also appear in other groups within this table.												
20												

Рисунок 1 — Примеры компоновки таблицы из коллекций: (а) DeEx, (б) SAUS, (в) CIUS. Цвета отображают класс ячеек

ПРИЛОЖЕНИЕ В

attributes	data	header	metadata	derived	notes			
# 0	1	2	3	4	5	6	7	
0	Secured Property Tax	2015.0	Prior Year Tax Bill. N					
1	Parcel	Tax Rate Area	Assessment Year	Roll Year	Installment 1	Installment 2	Total	
2	035-320-470	12-001	2015	2015	General Tax	107803.88	107803.88	\$215,607.76
3	Owner Address				Total Special Charge	155.02	155.02	\$310.04
4	*Name private per C				Total Taxes	107958.90	107958.90	\$215,917.80
5	1875 S BASCOM AV				Penalty + Cost + Fee	0.00	0.00	\$0.00
6	CAMPBELL CA 9500				Total Amount	\$107,958.90	\$107,958.90	\$215,917.80
7	Property Location				Due Date Late After	Nov 01, 2015 Dec 10	Feb 01, 2016 Apr 10,	
8	2000 S DELAWARE				PAID DATE	DEC 07, 2015	MAR 15, 2016	
9	SAN MATEO				Detail Special Charge	Phone Contact	Amount	
10	Values				SMC Mosquito Abate	(800) 273-5167	3.74	
11	Improvements	13020000			NPDES Storm Drain	(650) 363-4100	1.72	
12	Land	6196378			More Special Charge			
13	Exemptions				Composite Rate	1.122	Penalty Rate	10.0%
14	Net value	\$19,216,378						
15	Legal Description	PARCEL B PARCEL						
16	Be aware that during							

Рисунок 1 — Пример определения класса ячеек нейронным классификатором у таблицы сложной структуры

attributes	data	header	metadata	derived	notes
# 0	1	2	3		
0	WARD	PARCEL NO.	BILL NUMBER	BANK NO.	
1	03	04406-109	998	13	
2	LOCATION				
3	160 FEDERAL ST G-54				

# 0	1	2	3	
0	Total Full Valuation	64000.0	Net Tax & Spec. Assmnt. Due	793.6
1	Residential Exemption	0.0	Preliminary Overdue	0.0
2	Total Taxable Valuation	64000.0	1St Tax Payments Due By 02/01/2019	396.8
3	Total Tax & Spec Assmnt. Due	1600.0	2Nd Tax Payments Due By 05/01/2019	396.8
4	Personal Exemptions	0.0	Tax Due	396.8
5	Payments To Date/Credits	806.4	Real Estate Amount	396.8
6	Bid Balance Due	13.07		
7	Total Due Pay By 02/01/2019	409.87		

Рисунок 2 — Пример определения класса ячеек нейронным классификатором у таблиц, принадлежащих разделенному классу

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Lehmberg, O., Ritze, D., Meusel, R., & Bizer, C. (2016). A Large Public Corpus of Web Tables containing Time and Context Metadata. *Proceedings of the 25th International Conference Companion on World Wide Web*.
2. Cafarella, M.J., Halevy, A.Y., Lee, H., Madhavan, J., Yu, C., Wang, D.Z., & Wu, E. (2018). Ten Years of WebTables. *Proc. VLDB Endow.*, 11, 2140-2149.
3. Wang, Y. and J. Hu (2002). A machine learning based approach for table detection on the web. In Proceedings of the 11th International Conference on the World Wide Web, pp. 242–250.
4. Crestan, E., & Pantel, P. (2011). Web-scale table census and classification. *Web Search and Data Mining*.
5. Lautert, L.R., Scheidt, M.M., & Dorneles, C.F. (2013). Web table taxonomy and formalization. *SIGMOD Rec.*, 42, 28-33.
6. Milosevic, N., Gregson, C., Hernandez, R., & Nenadic, G. (2016). Disentangling the Structure of Tables in Scientific Literature. *International Conference on Applications of Natural Language to Data Bases*. // // research [Electronic resource]. – 2016. – Mode of access: https://www.research.manchester.ac.uk/portal/files/41051279/Disentangling_the_Structure_of_Tables_in_Scientific_Literature.pdf – Date of access: 10.02.2022.
7. Lehmberg, O., Ritze, D., Meusel, R., & Bizer, C. (2016). A Large Public Corpus of Web Tables containing Time and Context Metadata. *Proceedings of the 25th International Conference Companion on World Wide Web*. // webdatacommons [Electronic resource]. – 2015. – Mode of access: <http://webdatacommons.org/webtables/2015/downloadInstructions.html> – Date of access: 30.03.2023.
8. Roldán, J.C., Jiménez, P., & Corchuelo, R. (2019). On extracting data from tables that are encoded using HTML. *Knowl. Based Syst.*, 190, 105157.
9. Silva, A.C., Jorge, A.M., & Torgo, L. (2006). Design of an end-to-end method to extract information from tables. *International Journal of Document Analysis and Recognition (IJ DAR)*, 8, 144-171.
10. Zhang, S., & Balog, K. (2020). Web Table Extraction, Retrieval, and Augmentation: A Survey. *ACM Trans. Intell. Syst. Technol.*, 11, 13:1-13:35.

11. Efthymiou, V., Hassanzadeh, O., Rodriguez-Muro, M., & Christophides, V. (2017). Matching Web Tables with Knowledge Base Entities: From Entity Lookups to Entity Embeddings. *International Workshop on the Semantic Web*. // homepages [Electronic resource]. – 2017. – Mode of access: <https://homepages.tuni.fi/konstantinos.stefanidis/docs/er18/efthymiou.pdf> – Date of access: 30.03.2023.
12. Cafarella, M.J., Halevy, A.Y., Zhang, Y., Wang, D.Z., & Wu, E. (2008). Uncovering the Relational Web. *International Workshop on the Web and Databases*.
13. Eberius, J., Braunschweig, K., Hentsch, M., Thiele, M., Ahmadov, A., & Lehner, W. (2015). Building the Dresden Web Table Corpus: A Classification Approach. *2015 IEEE/ACM 2nd International Symposium on Big Data Computing (BDC)*, 41-50.
14. Nishida, K., Sadamitsu, K., Higashinaka, R., & Matsuo, Y. (2017). Understanding the Semantic Structures of Tables with a Hybrid Deep Neural Network Architecture. *AAAI Conference on Artificial Intelligence*.
15. Habibi, Maryam & Starlinger, Johannes & Leser, Ulf. (2020). DeepTable: a permutation invariant neural network for table orientation classification. *Data Mining and Knowledge Discovery*. 34. 1-21. 10.1007/s10618-020-00711-x.
16. Braunschweig, K. (2015). Recovering the Semantics of Tabular Web Data.
17. Ghasemi-Gol, M., & Szekely, P.A. (2018). TabVec: Table Vectors for Classification of Web Tables. *ArXiv, abs/1802.06290*.
18. Chen, Z., & Cafarella, M.J. (2013). Automatic web spreadsheet data extraction.
19. Ghasemi-Gol, M., Pujara, J., & Szekely, P.A. (2019). Tabular Cell Classification Using Pre-Trained Cell Embeddings. *2019 IEEE International Conference on Data Mining (ICDM)*, 230-239.
20. Wang, Z., Dong, H., Jia, R., Li, J., Fu, Z., Han, S., & Zhang, D. (2020). TUTA: Tree-based Transformers for Generally Structured Table Pre-training. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.
21. Du, L., Gao, F., Chen, X., Jia, R., Wang, J., Han, S., & Zhang, D. (2021). TabularNet: A Neural Network Architecture for Understanding Semantic Structures of Tabular Data. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.
22. sair.synerise [Electronic resource]. – 2022. – Mode of access: <https://sair.synerise.com/fourier-feature-encoding/> – Date of access: 21.03.2023.