ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

Национальный исследовательский университет «Высшая школа экономики»

Факультет гуманитарных наук

Образовательная программа «Фундаментальная и компьютерная лингвистика»

КУРСОВАЯ РАБОТА

На тему «Автоматическое извлечение основных грамматических параметров языка из грамматик»

Тема на английском "Linguistic Features Extraction from Reference Grammars"

Студентка 2 курса
группы №202

Козлова Екатерина Руслановна

Научный руководитель
Толдова Светлана Юрьевна
доцент Школы лингвистики Факультета гуманитарных наук

Научный консультант
Сериков Олег Алексеевич

Москва, 2022 г.

Оглавление

1. Введение

По данным Ethnologue (Eberhard et al 2022) на 2022 год в мире используется 7151 язык (кроме того, не стоит забывать о мертвых, также составляющих большую выборку). Однако для того, чтобы постулировать существование какого-либо идиома, недостаточно придумать название и поставить точку на карте: его еще нужно описать. Ведь одна из главных задач лингвистики – сохранить языковой материал.

В 1-м тысячелетии до н. э. сформировались первые лингвистические традиции: индийская, античная европейская и китайская (Алпатов 2005: 9), – и была создана первая грамматика санскрита. С течением времени традиции академического описания языков менялись и становились все более унифицированными. Так, было популяризовано глоссирование (Interlinear Morphemic Glossing, IMG) – в идеале, поморфемное сопоставление одного языка другому с целью передать значение каждого аффикса первого идиома (Lehmann 1982: 1). Следующим этапом стали создание и конвенционализация Лейпцигских правил глоссирования (Leipzig Glossing Rules, LGR), структурирующих систему условных обозначений. Таким образом, для современных научных грамматик был разработан единый стандарт языкового описания, однако его повсеместное применение, как будет показано дальше, еще не полностью закрепилось.

Основные «реформы» грамматического описания, как было отмечено выше, коснулись структуры языковых примеров. Также в процессе работы над корпусом грамматик было замечено, что в более поздних работах существует тенденция к представлению языкового материала в табличном виде, чего не было в ранних трудах. Так возникла идея анализировать именно эти два аспекта описания идиома, которые можно объединить под общим названием «структурированные лингвистические данные» или СЛД.

По определению «Большой российской энциклопедии» у слова «грамматика» есть несколько основных значения:

— Грамматический строй (система) – совокупность закономерностей к.-л. языка […].
— Научное произведение, описывающее грамматич. строй языка [...] (БРЭ).

Целью этой работы является автоматический анализ «грамматики» в широком смысле этого слова. Структурированность информации облегчает как ее восприятие (Федосеева, Алексеева 2018), так и применение методов автоматической

обработки. Более того, в контексте СЛД почти не важно, на каком языке написана грамматика, поэтому для ее восприятия после обработки читателю с меньшей вероятностью нужен будет переводчик. С помощью автоматического анализа может быть упрощена типологическая работа и сокращено время поиска релевантной для читателя информации в грамматике. Также автоматическая обработка позволяет за несколько минут создать первичное морфологическое описание языка (так, наличие показателя DU означает присутствие в языке двойственного числа), что применимо, например, в таких интерактивных электронных лингвистических ресурсах как ранее упоминавшийся Ethnologue или The World Atlas of Language Structures (Dryer, Haspelmath 2013). Кроме того, извлечение структурированных лингвистических данных может помочь в изучении языка с той точки зрения, что для большинства проиллюстрированных в грамматике показателей можно посмотреть не только их внешнее представление, но и множество примеров использования.

Таким образом, целю этой работы можно назвать создание онлайн-ресурса, позволяющего производить автоматические распознавание и обработку структурированных лингвистических данных. Пользователь должен иметь возможность просто загрузить ту или иную грамматическую работу определенного типа, а остальные действия выполнит программа. Результатом ее работы станет создание интерактивной базы данных отглоссированных примеров, грамматических таблиц и соответствий «морфема-глосса» со ссылками на образцы применения из языкового материала.

Для создания проекта использовался массив из более чем двух тысяч грамматик разного типа и времени создания, написанных «о и на» различных языках мира. Это позволило оценить многообразие традиции языкового описания, а также «откалибровать» работу ресурса, понять проблемы, с которыми он сталкивается, и дальнейшие направления развития. Код программы написан на языке программирования Python с применением методов оптического распознавания символов (Optical Character Recognition, OCR) и считывания табличной информации из файла с расширением PDF.

Ссылка на репозиторий с полным кодом и материалами работы размещена по ссылке: https://github.com/KatiaKozlova/grammar-analysis.

2. Обзор литературы и технических ресурсов по теме

Идея оптического распознавания текста не нова: первый метод был изобретен и запатентован еще в 1929 г. австрийским ученым Густавом Таушеком. Однако позднее OCR стало одним из направлений, проиллюстрировавшим успешный «симбиоз» методов программирования и лингвистики. Ведь только благодаря последней стало возможно научить программу «принимать решения» о том, какой из возможных символов более вероятен в контексте, а итоговые тексты стали более точными. Сейчас оптическое распознание текста является одной из задач компьютерной лингвистики, и существуют ресурсы, разрабатывающие решения в этой области, например, FineReader компании ABBYY. Существует также несколько общедоступных OCR-библиотек, написанных для языка Python, самой известной из которых является Python-tesseract, справляющаяся даже с мультиязыковым (Smith et al. 2009) и гетерогенным (Shafait, Smith 2010) материалом, подающимся на вход. Также существуют отдельные библиотеки для оптического распознавания таблиц, такие как Camelot и tabula-py.

Автоматическая обработка глоссированной информации же встречается во многих лингвистических работах (Samardzic et al. 2015; Zhao et al. 2020; Moeller, Hulden 2018). Однако их задача обычно сводится к машинному обучению на одном корпусе текстов, существующем в онлайн-формате, с целью разработки автоматического способа глоссирования.

Если же говорить про анализ табличной информации в языковых целях, подобные исследования встречаются лишь в связи с документальной лингвистикой, но не вполне отвечают целям этой работы, опираясь скорее на экономико-статистические документы (Дюженко 1975).

Стоит сказать, что идея этой работы вдохновлена проектом DReaM (The Dictionary/Grammar Reading Machine), целью которого было оцифровать и сделать доступными для поиска более чем 40,000 грамматических описаний и публикаций. Результатом их работы стал корпус лингвистических текстов Korp (Borin et al. 2012), на базе которого было написана, например, публикация, связанная с синтаксическим парсингом описательных грамматик естественных языков (Virk et al. 2017). Отдельно нужно отметить работу, посвященную именно извлече-

нию типологической информации из грамматик (Rama, Wichmann 2019). Ее основной целью является нахождение кусков текста, ассоциирующихся с тем или иным языковым явлением. Основными методами работы являются обучение на базе данных описаний из WALS и поиск наименьшего векторного расстояния при помощи коллокаций, ключевых слов, маркеров частей речи и частотности групп.

3. Данные

3. 1. Глоссирование

Стоит отметить, что формат, определенный Лейпцигскими правилами глоссирования выглядит, например, таким образом:

(1) Hiya-x-mun hun jo-cu-hnu.
I-ABS-FO I come-1REC.PAST-DECL
'It is I who has arrived.' [Sparing-Chávez 2012: 68]

Исходя из него и некоторых других ограничений на поиск по тексту, был создан список основных требований к примерам для их соответствия правилам глоссирования, а также адекватного распознавания:

- наличие эксплицитной нумерации;
- использование LGR (в том числе принятых разделителей для поморфемного членения и больших букв латиницы для основной части грамматических показателей);
- маркирование границ свободного перевода при помощи кавычек или апострофов;
- возможно, переносы первых двух строк IMG (см. (2));
- возможно, наличие первой строки после нумерации без глоссирования (на письменности языка или с пояснениями) (см. (3));

(2) ho-tə ka kirsən, kaka bisnu, mama
dist-ON uncle Krishna, uncle Visnnu, f.in.law
jəsbir, sohmlo rəhi-ke-rə
Jasbir, three remain-PFV-3P
'There, Uncle Krishna, Uncle Vishnu, Father-in-Law Jasbir, three remained.' [Watters 2004: 198]

(3) ꉢꂾꌠꈀ。

nga mop su ox.

1P.SG old man DP

‘I am an old man now.’ [Gerner 2013: 159]

3. 2. Табличные данные

Единственным требованием для адекватного распознавания таблиц является наличие эксплицитных границ ячеек (см. Рисунок 1).

Рисунок 1. Пример распознавания таблицы с эксплицитными границами из грамматики [Fabre 2014: 133]

5.1.1.2.7. Morfología de los verbos de la segunda conjugación.

	RAÍCES VOCÁLICAS	
	PREFIJOS NO GLOT. (39)	PREFIJOS GLOT. (33)
1ª p.	xay- y- **	xa’y- * y- **
2ª p.	lht- Ø **	lht’- Ø **
3ª p.	t- nt- **	t’- nt’- **
1INC	sht-	sht’

Cuadro 16. Segunda conjugación: prefijos personales del modo realis (verbos que empiezan por una vocal)

5.1.1.2.7. Morfología de los verbos de la segunda conjugación.

	RAÍCES VOCÁLICAS	
	PREFIJOS NO GLOT. (39)	PREFIJOS GLOT. (33)
1ª p.	xay- y- **	xa’y- * y- **
2ª p.	lht- Ø **	lht’- Ø **
3ª p.	t- nt- **	t’- nt’- **
1INC	sht-	sht’

Cuadro 16. Segunda conjugación: prefijos personales del modo realis (verbos que empiezan por una vocal)

3. 3. Формат подачи данных

На вход пользователь подает распознаваемую грамматику в формате PDF с оговоренным выше типом СЛД. Для большей точности распознавания это должен быть машиночитаемый файл.

3. 4. «Обучающая» база данных

Тестирование и отладка работы программы проводились на базе данных, состоящей из более чем 2400 грамматик и словарей 1363 языков мира, хранящихся в открытом доступе на облачном хранилище MEGA[1]. Надо сказать, что ее стоит считать обучающей в широком смысле этого слова. Так, в ходе исследования, именно на ее основе были сделаны выводы про основные стратегии кодирования СЛД, отходящие от принятых Лейпцигских правил. Это могли быть различные способы нумерации подпунктов примеров и отделения номера от текста, наличие строки с частеречной принадлежностью слова или его вариантом на письменности языка оригинала внутри IMG и другие отступления. Благодаря этой базе данных также

[1] https://mega.nz/folder/x4VG3DRL#lqecF4q2ywojGLE0O8cu4A.

стало понятно, что процесс унификации грамматического материала в современных лингвистических работах еще отнюдь не закончен, поэтому LGR являются ориентиром, но, пока к сожалению, не единым стандартом. Была также создана и размечена репрезентативная подвыборка[2] из 1000 файлов, на основе которой была посчитана статистика валидности применения ресурса (см. Таблицу 1).

Таблица 1. Статистика по распознаванию 1000 случайных грамматик из «обучающей» базы данных.

Распознается ли?	Причина	Всего, шт.	Всего, %
да		153	20.3%
скорее да	*отсканирована и оцифрована* (4)	50	
нет	*не распознана* (5)	451	69.1%
	не академическая (6)	111	
	нет IMG (7)	129	
	другой формат IMG (8)	106	10.6%
всего		1000	100.0%

Стоит уточнить, что почти половина подвыборки (8) состояла из нераспознанных PDF-файлов, что делало их обработку невозможной. Также многие грамматики не подходили из-за формата СЛД, либо являясь учебными (6), либо не содержа отглоссированных примеров (7) (что является повсеместным для грамматик, написанных в первой половине 20 в. или раньше). Причиной невалидности автоматической обработки для оставшихся грамматик (5) являлось нарушение одного из требований, заданных в Части 3. 1. Кроме того, 5% грамматик (4), подходя по структуре, являлись отсканированными и оцифрованными, что делает результат работы более непредсказуемым (см. Часть 3. 3).

3. 5. Формат данных на выходе

Пользователь получает 2 таблицы в формате CSV:

— примеры, извлеченные из грамматики с информацией для каждого о его номере в грамматике (9); странице, где он встречается в файле (10); записи на языке оригинала (11); строке глоссирования (12); переводе на язык грамматики (13) (см. Таблицу 2);

— соответствия «глосса-морфема» с информацией для каждого о его обозначении в IMG (14); морфеме в языке грамматики (15); примерах

[2] https://docs.google.com/spreadsheets/d/1Hjfru6VSZWIyt2Gg6ZRRtvAeQrgmOM2GAi8LMT6whs0/.

(p. *номер_страницы* (*номер_примера*)) и таблицах (p. *номер_страницы* tab. (*номер_таблицы*)), где оно встречается (16) (см. Таблицу 3).

Таблица 2. Пример организации таблицы для примера (18) [Overall 2007: 495].

ID	Number Example (9)	Page (10)	Example (11)	Glossing (12)	Translation (13)
51	18	518[3]	ʃiiha aniˌ-a-ha-i wiˌ-tasa-nu-ʃa ta-wa-ka	very desire-IMPFV-1SG-DECL go:PFV-INTENT-1SG-UNCERT say+IMPFV-3-POLINT	‘is she saying “I really want to go”?’

Таблица 3. Пример организации таблицы для соответствия “*1SG-nu*” [Overall 2007].

ID	Gloss (14)	Affix (15)	Examples (16)
20	1SG	-nu-	p. 518 (18), p. 554 (119), p. 555 (121), p. 388 tab. (8.17), p. 414 tab. (9.4)

Также создается папка с вырезанными и сохраненными в формате JPEG таблицами с морфологическими показателями из грамматики, у каждой из которых есть распознанная версия, сохраненная в формате CSV. Таблицы имеют название в формате «*номер страницы_номер таблицы*: *название таблицы в грамматике*» (см. Рисунок 2 и Таблицу 4).

Рисунок 2 (слева), Таблица 4 (справа). Вырезанная и распознанная таблица «385_8. 15: First and second person markers on past declarative verbs» [Overall 2007: 362].

PERSON	SINGULAR	NON-SINGULAR
1	*-ha*	*-hi*
2	*-umɨ*	*-uhumɨ*

PERSON	SINGULAR	NON-SINGULAR
1	-ha	-hi
2	-umɨ	-uhumɨ

3. 6. Сайт

Посредством веб-интерфейса можно как загрузить свою PDF-грамматику, оговоренного выше формата, так и выбрать одну из тех, что были отобраны из подвыборки в Части 3. 4. После обработки готовые файлы (см. Таблицу 2 и 3) подгружаются на сайт: их можно просматривать со станицы или скачать ZIP-архивом на компьютер.

[3] Номером страницы является номер в файле, а не внутри грамматики, что также облегчает дальнейший поиск в электронном документе

4. Методы

4. 1. Этапы обработки

Программа делится на три основных части, некоторые из которых членятся на несколько блоков и вспомогательных функций.

4. 1. 1. Извлечение структурированных лингвистических данных

Структурированная информация извлекается постранично. Сначала в тексте каждого листа при помощи регулярных выражений ищутся примеры и названия таблиц. Если находятся последние, то при помощи отдельной функции производится поиск граничных линий самой большой таблицы на странице и ее вырезание. После этого происходит ее извлечение в формате Data Frame и поиск морфем внутри (по наличию разделителей, таких как тире, дефис, равно и др.). Если таблица считывается как программой как морфологическая (при наличии более двух аффиксов), она сохраняется в формате CSV и, ее вырезанная версия, – в JPEG.

Из каждого примера извлекается нумерация, принятая внутри данной грамматики (по возможности, с подуровневыми обозначениями), удаляются «лишние» начальные строки (см. (2)) и пунктуация. После чего текст делится на три части: строка (или строки) на языке оригинала, глоссирование и перевод, – и сохраняется как список под номером страницы и примера в словарь формата JSON.

4. 1. 2. Обработка морфологической информации

Основной задачей на этом уровне является максимальное извлечение морфологической информации из примеров и таблиц. Как было описано в Части 3. 1, считается, что на границе элементов в IMG должен стоять один из разделителей (тире, дефис, равно или др.). Поэтому мы ищем аффиксы как последовательность символов, до или после которой находится какой-то из вышеупомянутых знаков. Однако понятно, что примеры и таблицы являются очень разнородными источниками данных, поэтому и для выделения соответствий «морфема-глосса» существует две независимые функции.

Первая, используя JSON файл с примерами (Таблица 5 (17)), для каждого списка производит «чистку» первых двух его частей (строка оригинала и глосс) от лишних пробелов и знаков пунктуации (Таблица 5 (18)). После этого они бьются по разделителям (Таблица 5 (19)). Для случаев, где количество элементов в первых

двух строках совпадает (для Таблицы 5 (19) – 12 штук), происходит поиск глосс для морфологических показателей во второй (по заданному списку разрешенных символов, таких как большие буквы латиницы, цифры и некоторые знаки препинания) (Таблица 5 (20)). При нахождении подобных, им сопоставляются аффиксы из первой строки и сохраняются вместе с номером и страницей примера в словарь формата JSON (Таблица 5 (21)).

Таблица 5. Процесс обработки примера (127) из грамматики [Polinsky 2015: 30] (кавычки опущены).

JSON-файл (17)	127_35: [[ʕAl-ä hat'an-ƛ'o ħäƛi-ru-(ɬi-)ce ħaƛu-s ħuł nes-ä. , Ali-ERG Sunday-SUPER.ESS drink-PST.PTCP-NMLZ-EQUAT drink-PST.EVID yesterday DEM.I-ERG , 'Yesterday he drank as much as Ali did on Sunday']]
«Чистка» (18)	ʕAl-ä hat'an-ƛ'o ħäƛi-ru-ɬi-ce ħaƛu-s ħuł nes-ä, Ali-ERG Sunday-SUPER.ESS drink-PST.PTCP-NMLZ-EQUAT drink-PST.EVID yesterday DEM.I-ERG
Деление на элементы (19)	[ʕAl, ä, hat'an, ƛ'o, ħäƛi, ru, ɬi, ce, ħaƛu, s, ħuł, nes, ä] [Ali, ERG, Sunday, SUPER.ESS, drink, PST.PTCP, NMLZ, EQUAT, drink, PST.EVID, yesterday, DEM.I, ERG]
Морфология (20)	[ä, ƛ'o, ru, ɬi, ce, s, nes, ä] [ERG, SUPER.ESS, PST.PTCP, NMLZ, EQUAT, PST.EVID, DEM.I, ERG]
Итоговый JSON-файл (21)	[ERG: [{ä: [127_35]}], SUPER.ESS: [{ƛ'o: [127_35]}], PST.PTCP: [{ru: [127_35]}], NMLZ: [{ɬi: [127_35]}], EQUAT: [{ce: [127_35]}], PST.EVID: [{s: [127_35]}], DEM.I: [{nes: [127_35]}]]

Вторая функция используется при поиске морфологических аффиксов внутри таблицы. Для ее работы с сайта англоязычной Википедии скачивается немного расширенный список глосс из Лейпцигских правил глоссирования с расшифровками значения. На его основе автоматически создается словарь ключевых слов. Для каждой морфемы из таблицы проводится ее поиск в готовом словаре пар «морфема-глосса» из примеров. Также производится отбор слов из соответствующих ячейке с морфемой строки и столбца и их сопоставление ключевым словам из словаря LGR. В случае с арабскими и римскими цифрами признаки комбинируются из соответствующих строки и столбца. Так, например, в таблице ниже (см. Рисунок 3) морфеме *'-у-'* будет соответствовать глосса IISG и пример TAB. 347_1 (т.е. *таб. (1) на стр. 347*). В итоге, пары соответствий с указанием на номер таблицы также добавляются в JSON-словарь.

Рисунок 3. Вырезанная таблица «347_1: Agreement prefixes» [Polinsky 2015: 347].

	Singular	Plural
Gender I	Ø-	b-
Gender II	y-	r-
Gender III	b-	
Gender IV	r-	

4. 1. 3. Приведение информации к табличному формату

Происходит независимая обработка двух JSON-файлов: с примерами и соответствиями «морфема-глосса». Файл с примерами окончательно очищается от «лишних» знаков препинания и примеров (с несоответствием количества слов в глоссировании и языке оригинала) по тому же принципу, что обсуждался в Части 4. 2. 2. Оба файла переводятся в Data Frame и CSV-файлы (см. Таблицы 2 и 3, соответственно), которые и являются окончательным результатом работы программы.

4. 2. Сайт

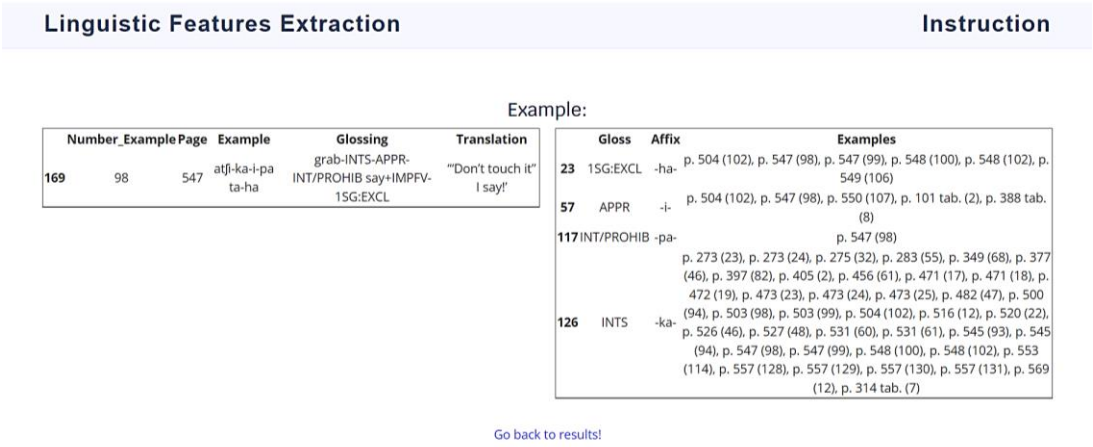
На стартовой странице сайта можно загрузить свою грамматику в формате PDF или выбрать одну любую анализируемую и отобранную в Части 3. 4 (см. Рисунок 4). Нажатием кнопки желаемый файл отправляется на сервер, начинается его обработка и отображается надпись с просьбой немного подождать. Когда заканчиваются распознавание и анализ, адрес автоматически перенаправляется на страницу, где находятся две таблицы с результатами. Каждое вхождение СЛД в базу данных «морфема-глосса» является ссылкой, нажав на которую можно увидеть отдельно эти картинку таблицы или глоссированный пример, а также все пары соответствий, что они иллюстрируют (Рисунок 5). Если вернуться обратно, внизу страницы с результатами также можно скачать интересующую пользователя распознанную информацию (по-отдельности или одним ZIP-архивом).

Верхнее меню веб-интерфейса позволяет перейти на стартовую страницу и на репозиторий GitHub[4] с описанием проекта и требований к подгружаемым грамматикам.

[4] https://github.com/KatiaKozlova/grammar-analysis.

Рисунок 4. Стартовая страница сайта.

Рисунок 5. Страница с результатами для одного примера (98) из грамматики [Overall 2007: 524].



4. 3. Библиотеки

Основной библиотекой, использующейся в коде программы, является pdfplumber. Она, по сравнению с другими, перечисленными в Части 2, обладает рядом преимуществ в условиях задач этой работы. Во-первых, в отличие от других ресурсов, pdfplumber способен в хорошем качестве одновременно распознавать как текст, так и табличную информацию, что ускоряет время работы. Во-вторых, он позволяет кастомизировать свои методы оптического распознавания: задавать физические границы, внутри которых применяется та или иная функция, обрезать страницу по ним и устанавливать эксплицитные рамки ячеек таблиц, – что повышает качество информации «на выходе». В-третьих, его скорость работы на порядок выше чем у других аналогичных библиотек.

У pdfplumber’а есть и свои недостатки, главным из которых является возможность распознавания лишь для машиночитаемого текста. Однако основной проблемой других библиотек, использующих более широкие возможности OCR, как говорилось выше, является значительное время, затрачиваемое распознавания. При создании данного ресурса же упор делался скорее на скорость, качество работы и возможность подстраиваться под запросы пользователя, чем на полноту возможностей распознавания. Это связано с тем, что программа, в первую очередь, позиционируется как удобное и быстрое средство первичной обработки. Именно поэтому выбор был сделан в пользу библиотеки pdfplumber.

Также устанавливаются Gdown (для скачивания файлов из Google Drive) и PyEnchant (созданная для проверки правописания, но применяющаяся для проверки существования слова в английском, французском, немецком или испанском языках). Последняя помогает отличить перенос слова на следующую строку от морфемы. Кроме того, используются также системные модули языка Python: re (регулярные выражения), json и csv (доступ к файлам соответствующих форматов), string (частотные операции со строками), os (работа с операционной системой), requests (работа с URL-адресами), – и программная библиотека pandas (обработка и анализ табличных данных).

В работе над сайтом использовались Flask (фреймворк для создания веб-приложений) библиотека Werkzeug (инструментарий для WSGI[5]), модули ZipFile (доступ к ZIP файлам) и shutil (файловые операции). Созданный на базе HTML, веб-интерфейс также включает части, написанные на CSS и JavaScript.

5. Оценка результатов и направления дальнейшей работы

Стоит уточнить, что созданный онлайн-ресурс, являясь интерактивным инструментом первичного анализа, лишь упрощает дальнейшую работу пользователя, но не является полноценным инструментом по сбору всей морфологической информации из документа. Так, например, соответствуя цели удобства пользователя, упор делался скорее на скорость работы. Кроме того, взяв в расчет создание ресурса «с нуля», основное предпочтение было отдано, если говорить в терминах

[5] Стандарт обмена данными между веб-сервером и веб-приложением.

машинного обучения, точности анализа, а не полноте обработки всевозможных входных данных.

В процессе создания ресурса, были отмечены его «слабые места» и, соответственно, способы дальнейшего усовершенствования. Так, некоторые потенциальные технические доработки описаны на странице проекта в GitHub. Помимо того, как было описано в Части 3, реализована поддержка данных лишь определенной структуры, что задает направление дальнейшей работы в сторону обработки информации, не подходящий под вышеописанный формат. Возможными решениями являются применение глубоких методов компьютерного зрения или создание базы данных разнородных СЛД и модели для дальнейшего машинного обучения на них. Помимо того, грамматики в большинстве своем (45,1% из подвыборки) не являются машиночитаемыми (см. Таблицу 1), что указывает на еще одно направление будущей работы: сделать возможным обработку и анализ еще нераспознанных грамматик. Пока что все потенциальные решения этой проблемы сильно замедляют работу ресурса, что влечет за собой поиск баланса между скоростью и качеством анализа. Также в дальнейшей перспективе рассматривается возможность объединения усовершенствованного ресурса с другими, автоматизирующими нахождение морфологической информации в самом тексте грамматик, что позволяло бы создать более полную базу данных.

6. Заключение

Созданный ресурс отвечает тем целям, которые были поставлены в начале этой работы. Он автоматизирует распознавание и обработку структурированных данных, таких как таблицы и глоссированные примеры, из лингвистических работ, соответствующих Лейпцигским правилам, и создает первичное морфологическое описание языка. Его практическое применение позволяет упростить и ускорить извлечение типологической информации для, по предположительным данным из Части 3. 4, примерно 2/3[6] современных научных машиночитаемых грамматик. С учетом направлений дальнейшего усовершенствования данного ресурса, можно сказать, что этот проект является лишь первым шагом в сторону полноценной автоматизации извлечения морфологических параметров языка из грамматических работ.

[6] 65,7% грамматик, если ориентироваться на данные подвыборки (см. Таблицу 1).

Стоит принять во внимание, что применение автоматического анализа, в том числе в языкознании, является достаточно актуальным и часто встречается в современных научных работах. Однако идея его использования в лингвистике для обработки большого числа разнородных структурированных источников информации, таких как «обучающая» база данных грамматик в этом проекте, появилась не так давно и во многом связана с ранее упомянутым проектом DReaM. Поэтому хотя, с одной стороны, новизна и актуальность этой работы не вызывают сомнений, с другой стороны, она покрывает лишь малую часть обширного поля будущей деятельности в данном направлении.

Литература

Алпатов 2005 — В. М. Алпатов. *История лингвистических учений*. М.: Языки славянской культуры, 2005.

БРЭ — С. А. Крылов. Грамматика // *Большая российская энциклопедия* (электронный ресурс), 2016. URL: https://bigenc.ru/linguistics/text/2375733.

Дюженко 1975 — Г. А. Дюженко. *Документальная лингвистика*. М.: Статистика, 1975.

Федосеева, Алексеева 2018 — Л. Н. Федосеева, Т. Е. Алексеева. Структурирование и визуализация информации как способ повышения эффективности учебного процесса (на материале английского языка) // *Теоретические и прикладные аспекты развития современной науки и образования*. Чебоксары: Негосударственное образовательное частное учреждение дополнительного профессионального образования "Экспертно-методический центр", 2018. С. 59-65.

Borin et al. 2012 — L. Borin, M. Forsberg, J. Roxendal. Korp – the corpus infrastructure of Språkbanken // *Proceedings of LREC 2012*. Istanbul: ELRA, 2012. P. 474-478.

DReaM — Sh. M. Virk, M. Forsberg, H. Hammarström. *DReaM: The Dictionary/Grammar Reading Machine*. URL: https://spraakbanken.gu.se/projekt/dream.

Dryer, Haspelmath 2013 — M. S. Dryer, M. Haspelmath (eds.). *The World Atlas of Language Structures Online*. Leipzig: Max Planck Institute for Evolutionary Anthropology, 2013. URL: http://wals.info.

Eberhard et al 2022 — D. M. Eberhard, G. F. Simons, Ch. D. Fennig (eds.). *Ethnologue: Languages of the World. Twenty-fifth edition.* Dallas: SIL International, 2022.

URL: https://www.ethnologue.com/.

Fabre 2014 — A. Fabre. *Estudio gramatical de la lengua Nivacle*. München: LINCOM GmbH, 2014.

Gerner 2013 — M. Gerner. *A Grammar of Nuosu*. Berlin, Boston: De Gruyter Mouton, 2013.

Lehmann 1982 — Ch. Lehmann. Directions for Interlinear Morphemic Translations // O. C. M. Fischer, M. Napoli (eds.). *Folia Linguistica 16(1-4)*, 1982. P. 199-224.

LGR — Leipzig Glossing Rules (электронный ресурс), 2015. URL: https://www.eva.mpg.de/lingua/resources/glossing-rules.php.

Moeller, Hulden 2018 — S. Moeller, M. Hulden. Automatic Glossing in a Low-Resource Setting for Language Documentation // J. L. Klavans (ed.). *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*. Santa Fe: Association for Computational Linguistics, 2018. P. 84-93.

Overall 2007 — S. E. Overall. *A Grammar of Aguaruna (Iiniá Chicham)*. Berlin, Boston: De Gruyter Mouton, 2007.

Polinsky 2015 — M. Polinsky. *Tsez Syntax: A Description*, 2015.

Rama, Wichmann 2019 —T. Rama, S. Wichmann. Towards unsupervised extraction of linguistic typological features from language descriptions // First Workshop on Typology for Polyglot NLP. Florence, 2019.

Samardzic et al. 2015 — T. Samardžić, R. Schikowski, S. Stoll. Automatic interlinear glossing as two-level sequence classification // K. Zervanou, M. van Erp, B. Alex (eds.). Proceedings of the 9th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH). Beijing: Association for Computational Linguistics, 2015. P. 68-72.

Shafait, Smith 2010 — F. Shafait, R. Smith. Table detection in heterogeneous documents // DAS '10: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems. NY: Association for Computing Machinery, 2010. P. 65-72.

Smith et al. 2009 — R. Smith, D. Antonova, D.-Sh. Lee. Adapting the Tesseract open source OCR engine for multilingual OCR // MOCR '09: Proceedings of the International Workshop on Multilingual OCR. NY: Association for Computing Machinery, 2009. P. 1-8.

Sparing-Chávez 2012 — M. Sparing-Chávez. *Aspects of Amahuaca Grammar. An Endangered Language of the Amazon Basin*. Dallas: SIL International, 2012.

Virk et al. 2017 — Sh. M. Virk, L. Borin, A. Saxena, H. Hammarström. Automatic Extraction of Typological Linguistic Features from Descriptive Grammars // K. Ekštein, V. Matoušek (eds.). *Text, Speech, and Dialogue. TSD 2017. Lecture Notes in Computer Science 10415*. Berlin: Springer, 2017. P. 111-119.

Watters 2004 — D. E. Watters. *A Grammar of Kham*. Cambridge: Cambridge University Press, 2004.

Zhao et al. 2020 — X. Zhao, S. Ozaki, A. Anastasopoulos, G. Neubig, L. Levin. Automatic Interlinear Glossing for Under-Resourced Languages Leveraging Translations // D. Scott, N. Bel, Ch. Zong (eds.). Proceedings of the 28th International Conference on Computational Linguistics. Barcelona: International Committee on Computational Linguistics, 2020. P. 5397-5408.

Технические ресурсы

Gdown — K. Wada. Gdown: Download a large file from Google Drive. URL: https://github.com/wkentaro/gdown.

Camelot — V. Mehta. Camelot: PDF Table Extraction for Humans. URL: https://camelot-py.readthedocs.io/.

FineReader — ABBYY FineReader PDF. URL: https://pdf.abbyy.com/.

MEGA — MEGA: Collection of descriptive grammars and pedagogical textbooks. URL: https://mega.nz/folder/x4VG3DRL#lqecF4q2ywojGLE0O8cu4A.

Pdfplumber — J. Singer-Vine, S. Jain. Pdfplumber. URL: https://github.com/jsvine/pdfplumber.

PyEnchant — D. Merejkowsky. PyEnchant. URL: https://pyenchant.github.io/pyenchant/.

Python-tesseract — S. Hoffstaetter. Python-tesseract. URL: https://github.com/madmaze/pytesseract.

Tabula-py — Tabula-py: Read tables in a PDF into DataFrame. URL: https://github.com/chezou/tabula-py.