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 «Автоматический морфологический анализ шугнанского языка»
Тема на английском: Automatic full morphology analysis for Shughni

Студент 2 курса
группы № 194
Меленченко Максим Глебович
(Ф.И.О.)

Научный руководитель
Ляшевская Ольга Николаевна
(Ф.И.О)
профессор
(должность, звание)

Москва, 2021 г.

Оглавление

1. Введение

Данная курсовая работа посвящена морфологическому анализу шугнанского языка. Целью работы является создание автоматического анализатора, который бы выдавал пользователю морфологический разбор каждого токена поданного текста. Такой анализатор может существенно помочь в лингвистических исследованиях. В задачи входят:

- разработка базовой схемы анализа;
- разработка дополнительных функций для отсеивания результатов согласно морфологическим ограничениям языка;
- разработка функции для распознавания «стяжённых» глагольных форм;
- разработка функции для решения проблем диалектной вариативности в написании и ошибок в записи текстов;
- разработка конвертера орфографий для решения проблемы графической вариативности;
- разработка веб-интерфейса в среде Flask;
- оценка эффективности результата с привлечением нескольких разобранных текстов.

В 2020 году в ВШЭ открылся проект «Компьютерные и лингвистические ресурсы для поддержки шугнанского языка» под руководством Е. В. Рахилиной[1]. В рамках проекта разрабатываются онлайн-словарь и корпус шугнанского языка, а также инструменты для автоматической обработки. В распоряжении проекта находятся материалы и тексты, полученные в том числе в ходе экспедиций в Таджикистан, которые помогают в разработке этих ресурсов. Создание системы автоматического морфологического анализа также является частью этого проекта. Анализатор предназначен для разбора текстов разного рода: экспедиционных устных текстов, полученных методами элицитации, печатной литературы

[1] Сердечно благодарю участников проекта: Ю. Макарова за организацию вычитки и разметки словаря Д. Карамшоева, разработку веб-сайта и ценные советы, Ф. Даниэль за неоценимую помощь в разработке анализатора, В. Плунгяна и А. Сергиенко за помощь в вопросах, связанных с шугнанским языком, Д. Новокшанова за предоставленные для тестирования тексты корпуса и помощь в их обработке, а также всех остальных участников проекта, которые участвовали в вычитке и разметке словаря, помогали советами и отзывами по работе анализатора.

и многих других, а также разного времени создания: от 1930-х годов до наших дней.

Далее в разделе «Шугнанский язык» представлено краткое описание грамматической системы шугнанского языка, необходимых для понимания работы, и явлений морфологии и графики, представляющих проблемы для разработки анализатора. Раздел «Данные» посвящён описанию материалов, с помощью которых разрабатывался и тестировался анализатор: это словарь Додхудо Карамшоева, обеспечивший лексическое наполнение, дополнительный словарь морфем, и тексты, использовавшиеся для оценки результата. В разделе «Методы» объясняется схема анализа, используемая программой, способы её реализации и дополнительные инструменты, улучшающие работу. В разделе «Оценка качества работы анализатора» представлены результаты оценки эффективности программы на материале нескольких текстов, а также проведён анализ некоторых показательных случаев ошибок её работы.

2. Шугнанский язык

Шугнанский язык — один из памирских языков, принадлежащих иранской группе индоевропейской языковой семьи. Он распространён в Афганистане и Таджикистане, в бассейне реки Пяндж, по которой установлена граница между государствами (Карамшоев 1988: 5—6). Число носителей, по разным оценкам, достигает от 80 до 100 тысяч человек (Edelman, Dodykhudoeva 2009: 788; Ethnologue).

Именная морфология в большей степени агглютинативная (Edelman 2009: 792). Существительное может иметь один или несколько суффиксов, реже — префикс, а также присоединять клитики, например:

(1) *di* *qišloq=and=ga* *yi* *ɣulā*
DEM2.M.SG.OBL кишлак=LOC=больше один большой
boǰak-zor *vuδ-ǰ*
орех-PLACE быть-PF
‘В этом кишлаке был большой орешник’ [из текста *Duzd yůrx̌ak*]

В роли клитик к имени могут присоединяться эмфатические частицы или маркеры лица субъекта клаузы, а также многочисленные послелоги. Послелоги таким образом сливаются с именем и отчасти выполняют функции падежей, утраченных в процессе развития языка. На письме клитики могут присоединяться к словам, записываться через дефис или знак равенства или отделяться от своей основы пробелом. Категория числа маркируется суффиксами: обычно это суффикс *-en*, но у ограниченного ряда имён используются другие исконные суффиксы. Различаются два числа: единственное и двойственное, других словоизменительных категорий у имён нет (Edelman, Dodykhudoeva 2009: 793—796).

Глаголы в шугнанском языке являются закрытым классом: в шугнанско-русском словаре Д. Карамшоева не более 650 глагольных статей. Для выражения сложных значений используются типичные для иранских языков сложные конструкции с именами, например: *ziv* ‘речь’ + *δêdow* ‘давать’ => *ziv δêdow* ‘уговаривать’ (Эдельман 1999: 240). Глагольные формы образуются от основ, которые могут различаться для разных форм времени, числа и рода. В этом состоит отличительная особенность шугнанского языка: основы для разных форм зачастую образуются нерегулярно. Попытка систематизации чередований, образующих разные основы, была проведена в (Муравьёва 1975), в результате чего было выявлено много небольших групп глаголов, в каждой из которых свои правила деривации. Такая интерпретация языкового материала не облегчает задачи морфологического анализа, поэтому эти глагольные формы далее рассматриваются как нерегулярные.

Глаголы имеют следующие базовые видовременные категории: настоящее время (презенс), прошедшее время (претерит), перфект, императив и инфинитив. Для регулярных глаголов от основы настоящего времени форма претерита образуется с помощью суффикса *-d*/*-t*, форма перфекта — с помощью суффикса *-ǰ*/*-č* (Edelman, Dodykhudoeva 2009: 797—798). У некоторых непереходных глаголов формы претерита и перфекта могут также различаться по роду и числу субъекта: например, глагол ‘стать’ имеет основу *saw-* в презенсе; в претерите — *su-* для мужского рода и *sa-* для женского и множественного числа; в перфекте — *suδ-* для мужского рода, *si-* для женского и *saδ-* для множественного числа.

Для форм перфекта женского рода используется особое окончание *-c* (например, *suδǰ — sic — saδǰ*). К перфектной основе может присоединяться суффикс *-at*, дающий форму плюсквамперфекта, например: *suδǰ-at-um* (стать.PF.M-PQP-1SG) (Edelman, Dodykhudoeva 2009: 800). Будущее время выражается формами презенса или специальными аналитическими конструкциями (Edelman, Dodykhudoeva 2009: 806).

Глагол в финитных формах изменяется по лицу и числу, для маркирования используются в настоящем времени суффиксы, в прошедших — схожие с ними по внешнему виду клитики. Клитики могут отделяться от своих глаголов и перемещаться к другому слову в клаузе. Глаголы также могут присоединять приставки, число которых ограничено; самые частые — приставки отрицательности *na-* (NEG) и *mā-* (PROH, запретительная: *mā-y-ad* ‘не приходи’) (Edelman, Dodykhudoeva 2009: 799).

Одна из особенностей употреблений глаголов шугнанского языка, заставляющая проводить дополнительный анализ токена, — стяжённые формы в настоящем времени. Формы настоящего времени, строящиеся по шаблону «основа + личное окончание», могут «стягиваться» специальным образом, при этом усекается конец основы: *anǰ-en* ← *anǰāv-en* (держать-3PL); *ma-m* ← *mar-um* (умереть-1SG). Стяжённые формы существуют только для форм 1SG, 2PL и 3PL, и только для основ, оканчивающихся в полной форме на кластер «гласная + согласная» (как *anǰāv-*). В случае форм 1SG усекаются последняя согласная основы и первая гласная личного окончания *-um*. В случае форм множественного числа ситуация другая: гласная и согласная конца основы усекаются, а окончание остаётся без изменений.

Анализ словоформ шугнанского языка осложняется некоторыми особенностями морфонологии. Во-первых, очень часто встречается сандхи — дополнительный [j], которая вставляется между морфемами на стыке гласных:

(2) *bād* *naxǰti-y-en* *tar* *māraka-y-en*
после уходить-[j]-3PL LOC собрание-[j]-PL
‘После этого ушла на собрание’ [из одного из текстов проекта]

Как показывает анализ практики глоссирования, в существующих глоссированиях чаще принято относить сандхи [j] к морфеме, дальней от корня, особенно в суффиксах: *māraka-yen*. Такой подход не очень систематичен и в некоторых случаях может запутать читателя глоссирования, поэтому анализатор воспринимает и выдаёт сандхи как отдельную морфему.

Другая проблема — выбор алломорфов *-d*/*-t* и *-ǰ*/*-č* в глагольных суффиксах. Этот вопрос на данный момент не изучен. По словарю можно наблюдать, что правила выбора зачастую не тривиальны и не интуитивны (например, ‘заставлять стоять’: настоящее время: *wiremb-*, перфект: *wiremb-č*/*wirem-ǰ*). В связи с этим анализатор на данный момент может распознавать как правильные в том числе формы с некорретными сочетаниями суффиксов (например, **wiremb-ǰ*) — предполагается, что таких ошибок в текстах крайне мало.

Шугнанский соседствует с родственным ему более крупным таджикским языком и испытал значительное влияние с его стороны. Многие таджикистанские шугнанцы владеют таджикским; он фактически является языком культуры, образования и делопроизводства. Поэтому речь шугнанцев изобилует таджикскими заимствованиями, которые в разной степени зафиксированы в словаре Карамшоева (Эдельман 1999: 225—227, 241). О глубине проникновения таджикизмов можно судить по шугнанским числительным: исконная система числительных активно заменяется новой, заимствованной из таджикского. В старой системе десятки образуются умножением: *cavor-δīs* ‘четыре десятка’, а в новой — супплетивно: *čil* ‘сорок’, из таджикского *чил* < *чиҳил* (Edelman, Dodykhudoeva 2009: 797).

Зачастую в текстах встречаются неосвоенные таджикизмы и русизмы. Например, в следующих примерах видны заимствования русского слова *технически* и таджикских *шӯравӣ* ‘советский’ (шугнанский аналог — *šūroyi*) и *ҷанг* ‘война’:

(3) ***Tehnicheski*** *misol ikam šič rāng ca nayatā* [из текста *Mama*]

(4) *ukumati* ***šūrāwi*** *vud wi dawrā* ***ǰang*** *dawrāyi* ***ǰang*** *vud* [2] [из текста *Bio2*]

Шугнанский язык не имеет устоявшейся графической традиции. Афганистанские тексты могут быть записаны с помощью арабицы, но такие тексты обычно не рассматриваются в московском проекте. Таджикистанские тексты могут быть написаны на кириллице или латинице, при этом конкретных вариантов кириллических и латинских алфавитов для шугнанского языка огромное множество. Латиница использовалась в образовании и печати в 1930-х годах; с 1990-х во многих текстах используется таджикская кириллица, адаптированная под шугнанскую фонологию (Каландаров 2020: 7—19). Существуют фонемы, для которых нет закрепившихся способов отображения в этих алфавитах, например, для записи звонкого велярного фрикативного согласного /ɣ/ могут использоваться следующие пары графем (соответственно заглавная и строчная): Ӻ/ӻ, Ɣ/ɣ, Ґ́/ҕ, Ґ/ґ̌, Ґ́/ґ̌, Г/г, Ұ/ұ, Ґ́ъ/ґ́ъ; для записи глухого зубного фрикативного /θ/ — Ө/ө, Ө/θ, Т́/т̌, Тъ/тъ, Ҍ/ҍ. Отдельная проблема связана с передачей печатных символов в Юникоде, где можно записать некоторые буквы с диакритическими знаками как цельный символ или как комбинацию отдельных буквы и знака. Хотя для обычного пользователя эти варианты выглядят одинаково, при работе с текстом (например, при поиске) могут возникнуть трудности.

В проекте используется особый алфавит на основе латиницы; все тексты проекта переводятся в него для унификации и облегчения анализа и чтения. Актуальную модификацию алфавита можно найти в Приложении или по ссылке: https://docs.google.com/spreadsheets/d/1x7PC3kBWLd1fKxCH_odgrKGxyO5ucSTE_vSTOdTdILs/edit?usp=sharing. Для проекта был отдельно разработан конвертер орфографий, который переводит тексты, записанные в любой известной кириллической или латинской орфографии, в этот алфавит. Из соображений единообразия все шугнанские слова и тексты в этой работе записаны с помощью проектного алфавита, если не указано иное.

[2] Как заметили В. Плунгян и А. Сергиенко, здесь заимствовано не просто слово *šūrāwi*, а целая конструкция: *ukūmat-i šūrāwi* (государство-IZ советский) ‘советская власть’. При этом слово *ukūmat* ‘государство’ — более раннее заимствование, и оно, в отличие от *šūrāwi*, есть в словаре Карамшоева.

В шугнанских текстах замечается значительная вариативность при написании некоторых букв. В одних и тех же словах в разных текстах могут различаться следующие пары графем и соответствующих фонем: a—ā, e—ê, i—ī, u—ū, u—ů, x—x̌ (то же верно и для соответствующих букв в кириллических алфавитах). Например, в одном из тестовых текстов встречается токен *rux̌δed-ard* (рассвет-LAT). В словаре Карамшоева, однако, корень ‘рассвет, утренняя заря’ указан как *рух-δêд* (в переводе в проектную орфографию *rux-δêd*). Буквы в парах e—ê и x—x̌ обозначают разные фонемы шугнанского языка. Причина таких расхождений иногда кроется в реальной языковой вариативности, в том числе диалектной, а иногда — в ошибках записи и распознавания текстов и словаря. Это обстоятельство представляет дополнительную проблему для разработки анализатора.

3. Данные

В этом разделе сначала описываются устройство словаря Д. Карамшоева, используемого в работе программы, и специальный скрипт, который оптимизирует словарь для анализатора. Затем описывается дополнительный словарь морфем, с которым также работает анализатор. В конце идёт описание текстов, которые использовались для оценки результатов работы.

3.1. Словарь Д. Карамшоева

Основным набором данных, обеспечивающим работу анализатора, является шугнанско-русский словарь памироведа Д. Карамшоева — самый большой русскоязычный словарь шугнанского языка на данный момент, размер которого составляет не менее 15 тысяч лексем (Карамшоев 1988). В рамках проекта летом 2020 года словарь был оцифрован, вычитан и размечен коллективными усилиями участников шугнанского проекта. Качество вычитки оказалось вполне удовлетворительным, особенно после череды дополнительных полуавтоматических исправлений, связанных с особенностями распознавания некоторых символов, используемых в словаре. Специализированная разметка помогла использовать труд Карамшоева для работы онлайн-словаря и анализатора.

Полученная версия словаря хранится в формате .json и имеет следующую структуру:


```
{
 "WORD": "охуӈ(ак)",
 "NOUN.PL": "охуӈнен/охуӈнхел",
 "LEX": "1) учитель",
 "VOC": "ОХУӇ(АК) (мн. охуӈнен, охуӈнхел) учитель; ш. да́ð-ен декуӈ, тама-йет охуӈ они крестьяне, вы учителя; б. wа̄ð охуӈнхе́л-ен ар сивишӑн тойд те учителя отправились на совещание."
},
{
 "NPST": "ви/брв. вай",
 "NPST.SH": "шб. вӣ/б. вê-",
 "NPST-3SG": "шб. вед/ш. вид",
 "PST": "вуд",
 "PST.F": "вад",
 "PF.M": "ш. вуðч̣/б. вӯðч̣",
 "PF.F/PL": "виц",
 "PF.PL": "ш. ваðч̣/б. воðч̣",
 "INF2": "видow",
 "LEX": "1) быть, существовать; быть в наличии, иметься 2) быть, находиться, присутствовать 3) быть, происходить, случаться 4) (тк. перфектная форма) 3-е л.: оказывается 5) (глагол-связка) быть, являться 6) употребляется как вспомогательный глагол в описательном преждепрошедшем перфекте",
 "IDIOMS": "ш. вуд на́-вуд, ш. вуðч̣, на́-вуðч̣, вӯðч̣ на́-вӯðч̣ (зачин сказок) было (ли), не было; жил(и)-был(и); дис ди вед, ш. дис ди вид раз (уж) так, если (уж) так; ме́θо вед (вуд); ру́зо вед (вуд) однажды, в один прекрасный день",
 "VOC": "шб. ВИ:ВУД, брв. ВАЙ:ВУД, ж. вад; сокр. шб. ВЙ-, б. ВЁ-; 3-е л. ед. ч. наст. шб. вед, ш. вид; перф. м. ш. вуðч̣, б. вӯðч̣, ж. виц, мн. ш. ваðч̣, б. воðч̣; инф. видow 1) быть, существовать; быть в наличии, иметься; <…>"
}
```


Словарь эксплицитно делит все лексемы на глаголы и не-глаголы. Глаголы хранятся особым образом: для них указываются формы NPST (основа презенса), PST (основа претерита), PF (основа перфекта), INF2 (инфинитив), IMPER (императив). Для некоторых лексем указываются и другие формы, если они образуются нерегулярно: NPST-3SG (3 лицо ед. ч. презенса), NPST.F (основа презенса для ж. р.), PST.F (основа претерита для ж. р.), PF.F/PL (основа перфекта для ж. р., а также для мн. ч., если отдельная основа не указана), PF.PL (основа перфекта только для мн. ч.). Также, в редких случаях, указаны стяжённые формы и соответствующие им полные: NPST-1SG, NPST.1SG.SH, NPST-2PL, NPST.2PL.SH, NPST-3PL, NPST.3PL.SH и краткая форма императива IMPER.SH.

У не-глаголов начальная форма слова указана в теге WORD. Для имён существительных могут указываться теги NOUN.GENDER (грамматический род), NOUN.PL (форма множественного числа), а также IDIOMS (идиомы, устойчивые

выражения с лексемой), CV (сложные глаголы, в именной части которых находится эта лексема). У всех лемм, и у глаголов, и у не-глаголов, указываются также теги LEX (лемма, значение слова) и VOC (полная словарная статья).

Одной из особенностей словаря Карамшоева является отсутствие имён собственных. В разделе «Оценка качества работы анализа» показано, что при анализе устных текстов это может явным образом ухудшить результативность анализатора.

3.2. Преобразование словаря

Для анализатора был разработан специальный скрипт, который обрабатывает .json-файл со словарём для нужд анализатора. Этот скрипт используется каждый раз, когда обновляется исходная версия словаря. Из статей удаляется содержимое тегов VOC и IDIOMS, ненужное для работы анализатора. С помощью регулярных выражений удаляются диалектные пометы («шх.», «брв.», «гнд.» и другие). Толкования слов, представленные одной строкой, делятся по специальным символам (запятым, точкам с запятым и слэшам), таким образом получается список строк, каждая из которых дсоответствует отдельному толкованию. С начала и конца каждой лексемы убираются знаки препинания (в основном это нужно для удаления дефисов и знаков равенства, которые в словаре могут обозначать границу морфемы). Лексемы с вариативностью, указанной в скобках, преобразуются в более удобный вид: текст вида *(х)авд* превращается в [*хавд*, *авд*]. Происходит конвертация орфографии шугнанских слов и основ. На каждом этапе удаляются пустые элементы структуры.

Скрипт выполняет и другие важные функции. В словаре для временных форм даны основы, то есть сочетание корня с маркером времени / рода, если такой есть (например, корень *vuδ* + суффикс перфекта *-ǰ*). Скрипт обрезает глагольные основы до корня: анализатор опознаёт маркер перфекта, и ему для работы требуется только сам корень. Если в словарной статье для существительного указаны формы множественного числа (под тегом NOUN.PL), то скрипт обрабатывает и их. Стандартная форма множественного числа выглядит как ‘корень-PL’ (PL обычно — суффикс *-en*), но бывают и супплетивные формы или стяжения. Если анализатор не находит корень

в форме множественного числа, то он записывает форму в поле NOUN.PL (например, у слова *buc* ‘дитя’ есть вариант множественного числа *bacen*). Если анализатор вычленяет корень в форме множественного числа, то он обрезает аффикс и добавляет его в поле PL.AFFIXES (например, у слова *dům̆od* ‘зять’ это поле содержит суффиксы *-en, -orǰ* и *-êrʒ*). Затем для этих аффиксов проверяется, нет ли среди форм в теге NOUN.PL таких, что оканчиваются на эти аффиксы. Если такие формы находятся, то их корень записывается в поле PL.STEMS. Таким образом в этом поле оказываются корни, которые используются только в формах множественного числа. Например, для словарной статьи вида:

"VOC": "ТĀНГЕҶ (мн. тāнгачен) житель селения Танг, тангец."

после работы скрипта в словаре окажутся в том числе поля WORD (*tāngeǰ*) и PL.STEMS (*tāngaǰ*).

Две словарные статьи, представленные в подразделе 3.1, после обработки имеют такой вид:


```
{
 "WORD": ["oxůnak", "oxůn"],
 "LEX": ["учитель"],
 "PL.AFFIXES": ["en", "xel"]
},
{
 "NPST": ["vi", "vay"],
 "NPST-3SG": ["ve", "vi"],
 "PST": ["vu"],
 "PST.F": ["va"],
 "PF.M": ["vuδ"],
 "PF.F/PL": ["vi"],
 "PF.PL": ["vaδ", "voδ"],
 "INF2": ["vi"],
 "LEX": ["быть"]
}
```


Помимо собственно полноценных слов и основ, в словаре Карамшоева встречаются также аффиксы и аффиксоиды[3]. Например:

ш. -РĀНГ, б. -РАНГ 1) часть слов с общим значением "вид", "род", "тип", напр.: ш. ар-рāнгаΘ, б. ар-рангаΘ, ш. ар-рангāҷ, б. ар-ран-гāҷ всякий, всякого рода; ш. йи-рāнг-(г)а, б. йи-ранг-(г)и другой, другого типа; ш. ца-рāнг, б. ца-ранг как, какой; почему; <…>

[3] Аффиксоиды — корни, в словосложениях приобретающие некоторые свойства аффиксов. Подробнее можно прочитать, например, в (Плунгян 2000: 86—87)

=УМ 2, =ЙУМ, =WУМ суффикс порядковых числительных, напр. wӯвдум седьмой; waxŭтум восьмой <…>

Словарь представляет шугнанское словообразование несколько избыточно: с одной стороны, для аффиксоидов выделены отдельные словарные статьи, а с другой, свои статьи выделены и для сложных слов, включающих в себя эти аффиксоиды. Это очень удобно для пользователя словаря, но анализатор не умеет автоматически различать статус словарной статьи, поэтому он будет считать аффиксоиды за корни. Так как аффиксы и аффиксоиды зачастую состоят из всего нескольких фонем, их легко можно найти в случайном токене, а значит, без предварительного отсеивания таких статей вывод анализатора будет заполнен некорректными разборами, в которых аффиксоид был распознан как корень. Например, без отсеивания в токене *mardum* ‘народ’ найдётся в том числе разбор *ma-rd-um* (PROH-LAT-1SG), где *ma-* — это глагольный префикс (PROH; отрицание, запрет), который анализатор считывает как корень. Поэтому скрипт также фильтрует список словарных статей. Фильтрация проходит на основе списка, который составлен вручную и хранится в отдельном .csv-файле. Если статья находится в этом списке, она не включается в версию словаря, предназначенную для анализатора.

Эта же фильтрующая функция служит и другой цели. Кроме словарных статей, которые нужно удалить, в этот список входят также и те, глоссирование которых (в теге LEX) нужно упростить или сократить. Это позволяет сделать выдачу более визуально удобной за счёт того, что часто встречающиеся лексемы получают вместо длинного описания значения более ёмкое, пригодное для глоссирования. Некоторые леммы заменяются на глоссы, что также облегчает совместимость с корпусом текстов, в котором используются такие глоссы. Несколько примеров изменений, которые производит скрипт (слева — фрагмент изначального значения, справа — изменённое), приведены в Таблице 1.

Таблица 1. Изменение лемм при обработке словаря

резать, убивать, закалывать животных <…>	резать
в значении личного местоимения: их, им, ими, (у) них и т. п. <…>	DEM2.PL.OBL

предлог 1. при косвенных дополнениях и обстоятельствах указывает на 1) направление движения, действия из какого-л. места <…>	EL

3.3. Словарь морфем

Специально для анализатора был создан автономный словарь шугнанских аффиксов и клитик, необходимых для распознавания. Он скомпилирован из нескольких источников: в первую очередь из вхождений морфем и отдельных лексем, встречающихся в словаре Д. Карамшоева, а также из нескольких описательных грамматик (Эдельман 1999; Edelman, Dodykhudoeva 2009) и специальной инструкции по глоссированию, разработанной участниками шугнанского проекта. Словарь клитик и аффиксов хранится в отдельном файле .csv-формата. В Таблице 2 можно увидеть примеры морфем из этого файла. Во втором столбце указан тип аффикса (префикс или суффикс), в третьем — его фонематическую запись, в четвёртом — глоссу, в пятом — тип основ, к которым аффикс может присоединяться (имена, глаголы или любые).

Таблица 2. Устройство словаря морфем

1	prefix	či	CONT	Noun
30	suffix	u	CONJ	—
9	prefix	par	REFC	Verb

3.4. Тексты

Для тестирования работы анализатора использовались четыре шугнанских текста, находящихся в распоряжении проекта, с условными названиями *Duzd yůrx̌ak, Bio3, Mama* и *Bio2* . Это четыре файла .json в определённом формате, который одновременно совместим с разрабатываемым корпусом шугнанского языка и имитирует вывод анализатора. *Duzd yůrx̌ak*, или ‘Медвежонок-вор’, — текст из сборника сказок. В проекте уже есть текстовое глоссирование этой сказки, которое было вручную переведено в нужный формат. Остальные три файла — это записи устной речи, полученные методом элицитации в результате экспедиций 2017—2019 годов. Это тексты из корпуса шугнанского

языка; они были получены от Д. Новокшанова, разработчика, а затем обработаны вручную. Длина текстов равна 145, 254, 264 и 326 токенов соответственно; в сумме 989 токенов (токены знаков препинаний не учитываются).

Корпус работает на платформе tsakorpus, свободно распространяемой платформе для создания корпусов, разработанной Т. А. Архангельским. В корпусе хранятся тексты в формате .json, которые могут также содержать морфологический разбор и/или метаданные. Веб-интерфейс корпусов tsakorpus написан на Python и Flask. Шугнанский корпус сейчас доступен на сайте проектов Школы лингвистики НИУ ВШЭ по ссылке https://linghub.ru/shughni_corpus/search. Устройство платформы tsakorpus подробно описано на её сайте, доступном по ссылке https://tsakorpus.readthedocs.io/en/latest.

Структура .json-файла с текстом выглядит следующим образом (на примере фрагмента одного из текстов):


```
{
 "meta": {
  "filename": "corpus\\shughni\\eaf\\shg_txt_DMO_RR_20180801_biography3.eaf",
  "title": "biography3",
  "author": "speaker",
  "год": 2018,
  "genre": "media-alligned"
 },
 "sentences": [
  {
   "text": "daδum lāk ču fukaθ xu naχtūydum tar nafaqā xu yima",
   "words": [
    {
     "wf": "daδum",
     "off_start": 0,
     "off_end": 5,
     "wtype": "word",
     "ana": [
      {"gloss": "после", "parts": "daδ", "gloss_index": "после{daδ}-"},
      {"gloss": "1SG", "parts": "um", "gloss_index": "1SG{um}-"}
     ],
     "next_word": 1,
     "sentence_index": 0
    },
   <…>
   ],
   "lang": 0,
   "meta": {"speaker": "ID_B3"},
   "para_alignment": [
    {"off_start": 0, "off_end": 51, "para_id": 73}
   ],
   "src_alignment": [
```

```
 {
  "off_start_src": "18.21",
  "off_end_src": "23.97",
  "true_off_start_src": 18.21,
  "off_start_sent": 0,
  "off_end_sent": 51,
  "mtype": "audio",
  "src_id": "18210_23970",
  "src": "shg_txt_DMO_RR_2018081_biography3-0-0.mp4"
 }
]
}
```


В теге meta указана метаинформация о тексте, а в теге sentences находится список словарей, в каждом из которых расположена информация об одном предложении. Для каждого предложения указываются полный текст (text), список слов (words) и дополнительная информация, обеспечивающая синхронизацию текста с аудиофайлами в корпусе. В теге words расположен список словарей, каждый из которых отвечает за конкретный токен. Для токена указываются внешняя форма (wf), тип (wtype; слово или знак препинания), разбор (ana) и дополнительные данные для работы корпуса. В теге ana хранятся словари, описывающие морфему и её глоссирование (parts, gloss), иногда также могут быть указаны необязательные характеристики: например, часть речи (gr.pos). Более общее и подробное описание структуры текстов tsakorpus можно найти здесь: https://tsakorpus.readthedocs.io/en/latest/data_model.html.

В качестве примеров в данной работе используются как фрагменты текстов, используемых при тестировании, так и других шугнанских текстов, находящихся в распоряжении проекта.

4. Методы

Код анализатора написан на языке Python. Первоначально была разработана версия для автономного использования через командную строку, а затем написанный код был встроен в общее Flask-приложение, обеспечивающее работу всему сайту шугнанского проекта.

При загрузке словаря особое внимание уделяется глаголам. Для работы анализатора требуется присутствие в статье всех базовых основ глагола: NPST, PST, PF, INF2, IMPER — между тем последние две формы зачастую в статьях

отсутствуют. Предполагается, что для тех словарных статей, в которых эти формы не указаны эксплицитно, они образуются регулярным образом. Функция, загружающая словарь, проверяет на «целостность» каждую глагольную статью и достраивает недостающие формы по регулярным правилам в случае необходимости.

Также перед работой с текстом загружается словарь морфем, хранящийся в отдельном .csv-файле. Для каждой морфемы в словаре указано, присоединяется она слева или справа от корня, а также где она может встречаться: только в глагольных, только в не-глагольных или в любых формах. Вся эта информация сохраняется при выгрузке словаря в память анализатора.

Анализатор принимает на вход текст, делит его на предложения итокенизирует. (Предварительное деление на предложения требуется для совместимости с корпусом, в котором фрагменты текстов представляются как предложения.) В токенах шугнанских текстов могут встречаться знаки препинания: дефисы, знаки равенства и, реже, точки, указывающие на границы морфем. Для токенизации используется Python-модуль nltk, который учитывает такие особенности и не считает внутрисловные знаки границами токенов.

Токены очищаются от лишних знаков препинания, которые могли остаться в нём из-за ошибок распознавания и разметки, и приводятся к строчному регистру. Некоторые знаки препинания остаются: например, дефис и знак равенства внутри токена могут означать границы между морфемами (обычно между клитиками и остальной частью токена), поэтому они не удаляются.

После очистки программа ищет совпадения между основами, собранными в словаре лексем, и токеном. Если какая-то основа находится, то сочетание лексемы (и соответствующей словарной статьи) и токена подвергается более детальному морфологическому анализу.

4.1. Морфологический анализ

Предварительно токен «разрезается» по границам морфем, обозначенным знаками равенства и дефиса. Такое деление облегчает процесс дальнейшего

анализа, а также предотвращает появление недопустимых разборов, в которых границы морфем не совпадают с местоположением таких разделительных знаков.

Затем из общего списка морфем составляется частный список для конкретного токена. Соответственно характеристикам формы, которую предполагает для токена анализатор, выбираются только те морфемы, которые могут потенциально встретиться в такой словоформе. Например, если анализатор предполагает, что в токене *naǎti-yēn* есть корень *naǎti* (выходить.NPST), то в частный список морфем не войдёт клитика-послелог *-ard* (LAT; латив), которая присоединяется только к существительным, но войдёт глагольная приставка *či-* (PROSP; проспектив).

Некоторые аффиксы не включены в словарь, они добавляются сразу в частный список в теле программы. Это личные окончания и суффиксы, маркирующие время глагола. Их добавлению в частный список предшествуют множественные условия и уточнения формы глагола. Так, в примере из предыдущего абзаца анализатор ищет форму презенса, поэтому в частный список войдут в том числе морфемы *-um* (1SG) и *-i* (2SG для презенса), но не войдут *-at* (2SG для претерита), *-ʒ* (маркер PF.F), *-ow* (SUP; супин, образующийся от основы инфинитива).

Затем программа пытается комбинаторно построить цепочки морфем из материала токена таким образом, чтобы одной из морфем была найденная основа, а окружающие её буквы образовывали подцепочки существующих морфем, зафиксированных в словаре клитик и аффиксов. К примеру, в токене *nalůvden* программа найдёт цепочку *na-lův-d-en* (NEG-говорить-PST-3PL). Теоретически также возможна цепочка **na-lův-d-en* (NEG-говорить-INF-3PL), но в частный список не могут попасть одновременно морфемы INF и 3PL, потому что анализатор знает, что форма инфинитива не имеет личных окончаний. Такие цепочки проходят через механизм отсеивания: программа предъявляет каждой цепочке некоторые требования, при несоответствии которым цепочка «бракуется», и итерация цикла прерывается. Функция sieve_sandhi проверяет, выполняются ли условия на появление эпентетической [j] между морфемами: она может стоять только между гласными. Например, тут будут отсеяны цепочки,

начинающиеся с неё (*y-am*), и цепочки, в которых она соседствует с согласными (*na-y-fĭč*). Функция sieve_verb_endings проверяет для цепочек, которым присвоена определённая глагольная форма, нашлись ли в них маркеры этих форм: например, отсеется цепочка *na-δoδ* (NEG-давать.PF), в которой использована основа перфекта *δoδ*, но нет суффикса перфекта *-ǰ*. Схожим образом функция sieve_personal_endings убирает цепочки глаголов в настоящем времени, в которых нет личных окончаний (например, *viwān* [отказываться от еды] — основа настоящего времени без окончания), а также те, в которых больше одного личного окончания. Другие функции проверяют, например, не повторяются ли аффиксы в цепочке (такое невозможно, за исключением некоторых отдельных случаев).

Так как в версии словаря, используемой в анализаторе, из глагольных лексем удалены маркеры форм, основы разных форм могут совпадать (например, основа глагола ‘сеять’ может выглядеть как *parwez* для презенса, претерита и перфекта). Можно подумать, что при таком подходе цепочки с одинаковыми основами будут неправомерно отсеиваться: программа забракует цепочку, выглядящую как *parwez-ǰ*, из-за того, что в ней нет личного окончания (а это могла бы быть форма претерита или перфекта, у которых личное окончание не обязательно). Однако на самом деле программа рассматривает каждую форму в словарной статье независимо и образует независимые цепочки. Поэтому в примере выше функции отсеивания принимают как минимум три цепочки, идентичные по структуре, но различающиеся по приписанной форме. После отсеивания сохранится только перфектная: цепочка, которая отмечена как форма презенса, не пройдёт проверку наличия личных окончаний, а у цепочки, которой приписана форма претерита, не обнаружится необходимого суффикса-маркера претерита.

Такие операции применяются отдельно к частям токена перед корнем и после корня. На выходе остаются цепочки морфем: цепочка проклитик и префиксов, затем корень (основа) и цепочка суффиксов и энклитик. Таких цепочек может быть достаточно большое количество. После этого они реорганизуются в общие цепочки для каждого токена. Например, из двух цепочек префиксов и двух цепочек суффиксов получится четыре конечных цепочки:

по одной на каждую возможную комбинацию. Каждая из таких цепочек будет состоять из n-го количества префиксов, корня и m-го количества суффиксов. После окончания стандартного анализа получается список возможных цепочек морфем; каждой морфеме присвоена некоторая глосса. Например, в токене *nalůvd-at* конечную морфему *-at* можно интерпретировать как личную энклитику 2SG и как клитику (ADD) ‘и’. На этом этапе произойдёт такая реструктуризация (структура упрощена для наглядности):

Таблица 3. Реструктуризация цепочек морфем: до и после

[[na (NEG)]], [[lův (говорить)]], [[d (PST), at (2SG)], [d (PST), at (ADD)]]	[[na (NEG), lův (говорить), d (PST), at (2SG)], [na (NEG), lův (говорить), d (PST), at (ADD)]]

4.2. Формат вывода

Вывод результата работы анализатора осуществляется в двух форматах. Первый — пользовательский, он выдаётся на веб-странице непосредственно после завершения работы. В нём каждый токен представлен как список потенциальных цепочек. Токены, для которых ни одного глоссирования не нашлось, тоже выводятся без разбора:

```
māš
# maš <1PL>
ukumat-i
# ukūmat <правительство; власть, власти>; i <ADJ>
# ukūmat <правительство; власть, власти>; i <SUBST>
# ukūmat <правительство; власть, власти>; i <I>
# ukūmat <правительство; власть, власти>; i <3SG>
# ukūmati <государственный>
šūrāwi
vud
# vu <быть, существовать; быть в наличии, иметься.PST.M>; d <PST>
```

Второй формат — .json-файлы, называемые output_raw.json и output.json. Файл output_raw.json — непосредственный результат работы программы, а output.json — результат дополнительной реструктуризации, в ходе которой

для каждого токена все списки цепочек морфем объединяются в один список. Это выглядит примерно так (структура упрощена для наглядности):

Таблица 4. Реструктуризация вывода анализатора в .json-формате: до и после

"ana": [[{"parts": "kin", "gloss": "делать.NPST"}, {"parts": "en", "gloss": "3PL"}], [{"parts": "kīn", "gloss": "ненависть"}, {"parts": "en", "gloss": "PL"}]]	"ana": [{"parts": "kin", "gloss": "делать.NPST"}, {"parts": "en", "gloss": "3PL"}, {"parts": "kīn", "gloss": "ненависть"}, {"parts": "en", "gloss": "PL"}]

Создание файла output.json необходимо для совместимости с корпусом. Анализ, приведённый в корпусе, не предполагает возможности множественных глоссирований. Поэтому результат работы анализатора с n глоссирований, каждое из которых состоит из морфем, упрощается до перечисления морфем без явного разделения на глоссирования[4].

Файл output.json можно получить, нажав на кнопку «Скачать json», появляющуюся на странице результатов анализа. Файл можно непосредственно загрузить в корпус, заполнив метаданные для доступа к соответствующему аудио.

4.3. Дополнительные функции

Одна из особенностей употреблений глаголов шугнанского языка, заставляющая проводить дополнительный анализ токена — стяжённые формы. Анализатор вынужден рассматривать их отдельно, потому что стандартный анализ не сможет найти полную основу из словаря в усечённой основе, присутствующей в токене. Для этого программа предварительно проверяет возможность интерпретации токена как стяжённой формы (отсеиваются токены с окончаниями, не соответствующими нужным личным окончаниям, а также

[4] Решение о таком формате было достигнуто вместе с Д. Новокшановым, который занимается корпусом.

глагольные основы, не оканчивающиеся на гласную и согласную). Затем, если все требования удовлетворены, основа усекается нужным образом и передаётся функции стандартного анализа с указанием на необходимость использования именно стяжённых личных окончаний при образовании цепочек.

Другие дополнительные функции связаны с настройками работы анализатора, которые пользователь может выбирать самостоятельно в веб-интерфейсе. Одна из таких настроек условно называется «игнорирование долгот гласных». Если эта настройка включена, все шугнанские морфемы из словаря морфем и корни из словаря Карамшоева будут специальным образом обработаны перед поиском совпадений. После обработки программа будет считать за один символ буквы в парах: a—ā, e—ê, ê—i, i—ī, u—ů, u—ū. Это позволяет решить проблемы с вариативностью в текстах, описанной в разделе «Шугнанский язык». Таким образом, например, в токене *buōr-ǰēv* (весна-TIME) найдётся словарный корень *buor* ‘весна’.

Другая опция, доступная пользователю, — «игнорирование дефисов». Как было описано в разделе 4.1, в некоторых текстах внутри токенов могут содержаться дефисы, знаки равенства или точки, разделяющие границы морфем — и анализатор учитывает их при делении токена на части. Если эта настройка включена, такие межморфемные знаки удаляются из токена на этапе очистки токена, то есть перед делением. Это позволяет найти правильный разбор для таких слов, как, например, *aft-um* (восемь-ORD) — это слово указано в словаре целиком как *aftum* ‘восьмой’.

Важнейшая настройка — «конвертация орфографий», о которой речь пойдёт в отдельном подразделе.

4.4. Конвертер орфографий

Для анализатора был разработан специальный конвертер орфографий. Он используется и автономно, для перевода текстов в принятый проектом алфавит (он доступен в отдельной вкладке на сайте), и как часть анализатора — в виде модуля Python, который импортируется в анализатор. Если пользователь включает настройку «Конвертация» при анализе, то текст будет предварительно конвертирован. Это рекомендуется делать во всех случаях, потому что словари

лексем и морфем, встроенные в анализатор, используют проектный алфавит, и без конвертации тексты в кириллической орфографии не будут анализироваться, а тексты в других орфографиях на основе латиницы будут анализироваться очень плохо.

Изначальная идея конвертера заключалась в том, чтобы построить таблицу соответствий разных алфавитов, для каждого текста определять алфавит автоматически по наличию определённых букв и заменять буквы одного алфавита на буквы другого последовательно. Однако выяснилось, что, несмотря на наличие некоторых более-менее устоявшихся орфографий, большое количество имеющихся текстов используют свои собственные кириллические или латинские алфавиты. Реализовать поддержку всех возможных систем письма не представлялось возможным. Вместе с тем отличия этих алфавитов почти всегда ограничиваются небольшим числом букв — это буквы, обозначающие шугнанские фонемы, отсутствующие в традиционных кириллице и латинице (например, /ɣ/, /ʊ:/, /dʑ/, /ɛ:/ и другие). Было решено реализовать другой подход: проходя по списку «неправильных» символов, конвертер заменяет все такие найденные символы на «правильные» соответствия. Например:

Таблица 5. Соответствия символов шугнанских орфографий

Другие алфавиты	Проектный алфавит
гъ, ґъ, ґ̌, ɣ̌, ɣ̱	ɣ̌
ӯ, ӯ, ӧ, ё, ӳ, ө	ů
ч, ǯ, ĵ, ǰ, љ	ǰ
ɛ, ɛ, ę, э, ə	ê

Как видно из примеров, такая замена решает не только проблемы собственно разнообразия письменностей, но и проблемы, связанные с отображением этих символов в электронном виде. Большинство текстов проекта — литература, записанная на бумаге, а затем уже отсканированная и распознанная, и тексты, полученные в ходе экспедиций. В обоих случаях авторов текстов по понятным причинам не интересовали проблемы отображения

символов в Unicode. Поэтому, например, фонема /dʑ/ в таких текстах может записываться визуально одинаковыми способами: «ǰ» и «ǰ», однако первый способ — это отдельно латинская «j» и гачек (U+006A и U+030C), а второй — неделимый символ (U+01F0). Другой пример: очень похожие, но отличающиеся в Unicode символы «ε» («греческая строчная буква эпсилон», U+03B5) и «ɛ» («латинская строчная буква открытая е», U+025B).

Таким образом, конвертер также решает проблему унификации, которая становится лёгкой благодаря отказу от таблиц (при использовании таблиц пришлось бы создавать отдельные «алфавиты» для каждой комбинации спорных случаев). Однако отказ от таблиц стал причиной появления другой проблемы. Некоторые символы могут обозначать разные фонемы в разных текстах. Обнаружено четыре таких случая: символы «ɣ» и «ɣ̌» (могут обозначать фонемы /ɣ̌/ и /ʁ/), «j» (может обозначать фонемы /j/ и /dʑ/, реже /dz/) и кластер «sh» (может обозначать фонему /ɕ/ или сочетание [sh]). При использовании таблиц алфавит определялся бы по наличию других символов, свойственных ему, и фонема, выражаемая символом, назначалась бы в соответствии с закреплённой ролью в конкретном алфавите. Для разрешения двойственности в конвертере используется примерно такой же механизм, только добавленный уже искусственно. Если программа обнаруживает неоднозначный символ, то она ищет в тексте другие символы, которые могут выражать потенциальные фонемы этого символа, и, если находит, считает «слот» этой фонемы уже занятым и не рассматривает её как вариант. Предполагается, разумеется, что в алфавитах каждого отдельного текста между фонемами и графемами существует взаимооднозначное соответствие: одна графема выражает одну фонему, и наоборот. Если какую-то неоднозначность разрешить не удаётся, то программа выбирает вариант «по умолчанию» — для каждой графемы существует такой вариант, который считается наиболее вероятным, если никаких дополнительных сведений о контексте нет.

(5a) *Xudowandard půnd **j**ītet at wi roh **y**en rost kinet.*

‘Приготовьте путь Господу, прямыми сделайте стези Ему’. [перевод Библии]

Например: на вход в конвертер подаётся фрагмент из Библии, программа обнаруживает в нём символ «j», который может иметь несколько фонетических значений. Однако в том же предложении есть символ «y», который обозначает фонему /j/. Слот этой фонемы оказывается занят, поэтому её присвоение букве «j» невозможно. Графем, которые обозначали бы /dʑ/ или /dz/, в предложении нет, поэтому из этих вариантов программа выбирает первый как более приоритетный. Результат работы конвертера будет выглядеть так:

(5b) *Xudowandard půnd **j̃**ītet at wi roh **y**en rost kinet.*

Возьмём другое предложение из того же текста и конвертируем его отдельно от предыдущего:

(6a) *Ba **j**aldī wi nům tar fuka **J**alīl tixīrm sut.*
‘И скоро разошлась о Нём молва по всей окрестности в Галилее’. [перевод Библии]

Здесь снова встречается «j», но других символов, которые могли бы обозначать его фонемы, нет. Поэтому будет выбран вариант /j/, более приоритетный, чем два других. Результат будет выглядеть так:

(6b) **Ba **y**aldī wi nům tar fuka **Y**alīl tixīrm sut.*

Первый пример — случай верной конвертации, второй — неверной. Так получилось потому, что текст во втором случае был недостаточно большого размера и в нём не нашлось необходимых для разрешения неоднозначности символов. Таким образом, по закону больших чисел, чем больше будет размер текста, тем меньше будет вероятность неверных интерпретаций символов.

В веб-интерфейсе конвертера предусмотрена возможность для пользователя самостоятельно определять соответствия проблемных графем. Это может быть удобно, если пользователь точно знает, какие символы каким соответствуют, а текст не слишком большого размера, чтобы полагаться на автораспознавание.

Помимо вышеперечисленного, конвертер решает ещё одну проблему, связанную с графикой. Эта проблема особенно остро встала после распознавания словаря Д. Карамшоева. Выяснилось, что во многих случаях там, где в тексте

словаря подразумевается кириллическая буква, программа распознавания увидела похожую на неё латинскую, или наоборот (например, приняла кириллическую «р» за латинскую «p»). Такое бывает и с другими текстами. Конвертер предварительно пытается определить общий тип алфавита по наличию специфичных букв и на этой основе заранее исправить возможные ошибки распознавания. Например, если во фрагменте из словаря *ту пуц жêхĭт**o**w-ирд нист* ‘твой сын бегает плохо’ конвертер обнаружит латинскую «o», то он определит, что тип алфавита — кириллица, по буквам «ц», «т», «д» и «ж», и заменит её на кириллическую.

Стоит напомнить, что для записи текстов шугнанцы используют и третий тип алфавита — арабский. К сожалению, арабица не поддерживается конвертером на данном этапе. Это связано не только с очевидными сложностями переноса текстов на арабице в фонематические, но и с тем, что арабица используется в основном в Афганистане, а московский проект изучает таджикский шугнанский и редко работает с текстами, записанными арабской графикой.

4.5. Веб-реализация

Анализатор доступен на сайте проекта «Компьютерные ресурсы для изучения шугнанского языка» по ссылке http://karamshoev.pythonanywhere.com. Веб-интерфейс конвертера и анализатора для сайта написан совместно с Ю. Макаровым, секретарём проекта, ответственным за распознавание и разметку словаря Карамшоева и главным разработчиком сайта. Сайт работает на Flask-приложении. Чтобы проанализировать текст, пользователю нужно открыть вкладку «Морфологический анализ», при необходимости выбрать нужные настройки, затем ввести текст в специальное поле или загрузить файл в формате .txt и нажать кнопку «Анализировать». После анализа страница обновится, и ниже кнопки «Анализировать» появится поле с результатами. В этом поле будут выводиться найденные глоссирования в пользовательском формате, описанном выше. Нажав на кнопку «Скачать json», можно также скачать исходный файл output.json, совместимый с форматом корпуса. Чуть ниже вывода указывается среднее время анализа на один токен текста.

5. Оценка качества работы анализатора

Результатом работы являются работающий морфологический анализатор шугнанского языка, доступный онлайн, и конвертер орфографий. По ссылке https://github.com/maxmerben/shughni_resources можно скачать автономную версию веб-сайта со словарём, конвертером и анализатором, которую можно запустить с собственного устройства со средой Flask. В этом разделе пойдёт речь об оценке эффективности работы анализатора.

5.1. Количественный анализ

Для оценки эффективности было проведено тестирование на четырёх текстах, разобранных в проекте. Перед тестированием они были дополнительно обработаны; обработка заключалась в унификации глосс из словаря Карамшоева и глосс из текста. Например, в одном из текстов встречается токен с корнем *mawlo*, который в исходном тексте глоссируется как ‘господь’, а в словаре Карамшоева (и, соответственно, в выводе анализатора) — ‘господь, бог’. Ручная обработка заключалась в приведении таких формально разных, но де-факто одинаковых выводов к одному стандарту. Затем специальный скрипт сравнивал два файла, вывод анализатора и обработанное корректное глоссирование, и подсчитывал метрики (см. Таблицу 6 и Диаграмму 1).

Таблица 6. Метрики тестирования работы анализатора

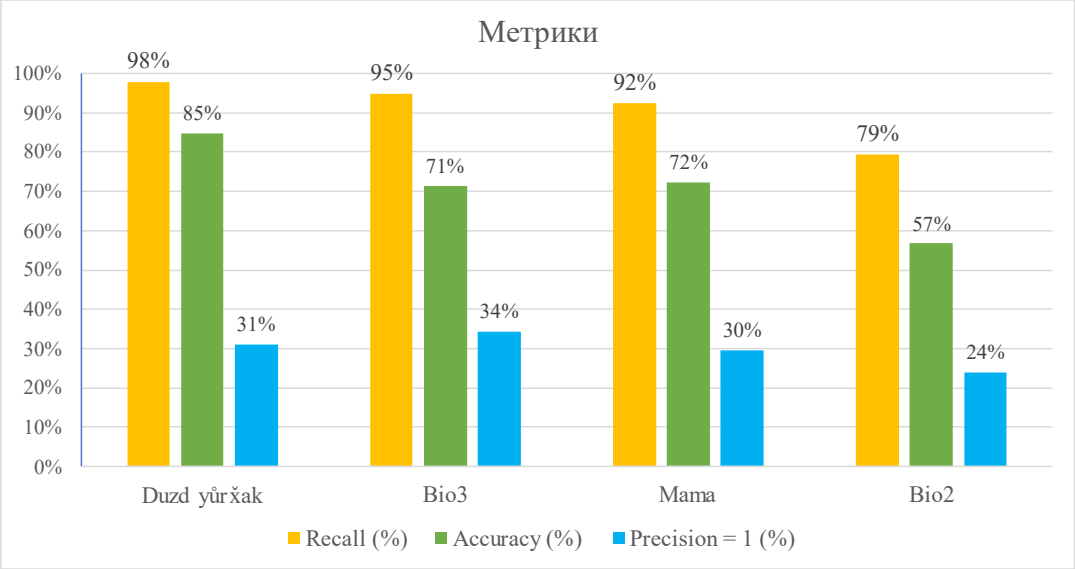
	Duzd yůrx̌ak	*Bio3*	*Mama*	*Bio2*	все тексты
Количество токенов	**145**	**254**	**264**	**326**	**989**
Recall	**142**	**241**	**244**	**259**	**886**
Recall (%)	98%	95%	92%	79%	90%
Accuracy	**123**	**181**	**191**	**185**	**680**
Accuracy (%)	85%	71%	72%	57%	69%
Precision=1	**45**	**87**	**78**	**78**	**288**
Precision=1 (%)	31%	34%	30%	24%	29%
Mean precision	0,5	0,48	0,45	0,36	0,45
Median precision	0,5	0,5	0,33	0,22	0,39
F-мера	66%	64%	61%	50%	60%

В строке «Количество токенов» указано количество токенов для каждого текста, за исключением токенов пунктуации (tex, у которых поле wtype в .json-структуре заполнено значением punc, а не word), без удаления повторяющихся токенов. В строках Recall и Recall (%) указаны количество токенов, для которых анализатор подобрал хотя бы один разбор, вне зависимости от корректности, и доля таких токенов в тексте. В строках Accuracy и Accuracy (%) указаны количество токенов, для которых среди всех разборов нашёлся правильный, и их доля среди всех токенов. Метрика Precision рассчитывается для каждого токена как отношение количества правильных разборов в выводе анализатора к количеству всех найденных анализатором разборов для этого токена (если ни одного разбора не предложено, precision считается за 0). В строках Precision=1 и Precision=1 (%) указаны количество и доля токенов, для которых эта метрика равна 1, то есть анализатор предложил один разбор, и он оказался правильным. (Это прототипически идеальная ситуация: если процент таких токенов будет стремиться к 100%, то исчезнет необходимость в дополнительной обработке вывода анализатора перед тем, как его можно загрузить в корпус.) В строках Mean precision и Median precision указаны соответственно среднее и медианное значение этой метрики для токенов каждого текста. F-мера, объединяющая метрики Precision и Recall, рассчитывается по следующей формуле:

$$\frac{2 \times Mean\,precision \times Recall}{Mean\,precision + Recall}$$

Нужно пояснить способ вычисления Precision. Оценка производилась сравнением с оригинальным глоссированием текста, произведённым участниками проекта. Между тем для многих токенов теоретически может быть несколько вариантов глоссирования. Например, вне контекста *vuδ-ǰ-at* можно разобрать как видеть-PF-2SG, видеть-PF-ADD и видеть-PF-PQP. Однако при сравнении только один, исходный вариант считается правильным. Поэтому если для токена *vuδ-ǰ-at* выдаются эти три разбора, верным будет считаться только тот, что в ручном разборе, и Precision токена будет равна 33 %.

Диаграмма 1. Recall, Accuracy и Precision=1 в тестовых текстах



Текст *Duzd yůrňak* использовался для тестирования анализатора на протяжении нескольких месяцев, поэтому некоторые проблемы, возникшие при анализе этого текста, были решены до проведения финального тестирования. Этим можно объяснить высокое значение Accuracy для этого текста. У текста Bio2, напротив, сравнительно низкие показатели всех метрик. По-видимому, низкие значения Accuracy и Precision являются следствием низкого значения Recall: в тексте много разговорных выражений (например, *ǰoni man, arodi man* ‘дорогой мой’), имён собственных (*Důlatšo, Germāniya, Đāx̌t* и другие), а также таджикизмов и числительных, которые анализатору неизвестны. Эти проблемы подробнее рассматриваются в следующем подразделе.

Скорость работы анализатора различается на несколько миллисекунд при каждом запуске — это зависит от нагрузки на вычислительные мощности машины. Средняя скорость анализа, замеренная при тестировании текстов, — 0,26 с / токен. Эта средняя скорость вычисляется как время от нажатия кнопки «Скачать json» в интерфейсе анализатора до вывода результатов пользователю, разделённое на количество токенов (токены пунктуации не учитываются).

5.2. Качественный анализ

Можно выделить несколько основных причин, по которым метрика Accuracy в текстах меньше 100%, то есть для большого числа токенов анализатор не предложил верного разбора.

Некоторые шугнанские лексемы, встречающиеся в текстах, отсутствуют в словаре Карамшоева. Например, токен *tadbīr-qati* содержит корень *tadbīr* ‘хитрость’, который не находится в словаре. Местоимённый корень *ed* в словаре встречается только в составе других слов (например, *k-ed* ‘тут’), но не имеет своей словарной статьи, из-за чего не распознаются такие токены, как *ik-tar-ed* (EMPH-EQ-DEM2). В текстах также есть морфемы, не добавленные в словарь морфем: например, из исследованных источников ничего не известно о суффиксе *-kunůn*, встречающемся в токене *xuši-kunůn* (радость-OWN). Кроме того, в текстах встречаются неосвоенные русизмы или таджикизмы, которые отсутствуют в словаре: к примеру, *tufli* ‘туфли’, *qaššoq* ‘бедный’ (ср. тадж. *қашшоқ*). Случается, что такие заимствования не распознаются из-за неустойчивости традиции записи этих слов. Поэтому, например, в токене *avtomobīli* не находится имеющийся в словаре корень *aftamubīl* ‘автомобиль’.

Наконец, результаты показывают, что на данный момент не все необходимые значения лексем были добавлены в словарь. В тексте *Duzd yůrx̌ak* всего 22 токена, для которых не был найден верный разбор, из них 13 — повторения местоимения *di* (DEM2.SG.OBL). В словаре Карамшоева несколько значений слова *di*, в числе которых и корректное, объединены в одну словарную статью. Анализатор нашёл эту словарную статью, но в выводе выдал неверное значение: ‘когда; в то время, как; как только; пока; до тех пор, пока’.

Зачастую в словаре отсутствуют имена собственные, в том числе географические названия, специфические для ареала распространения шугнанского. Там нет слова Памир (есть только прилагательное *pomeri* ‘памирский’), поэтому токен *Pomīr* остался нераспознанным. Нет в словаре и переводов для названий важных населённых пунктов шугнанского Таджикистана — города Хорог и села Шахдара (*Xaraɣ*, *Šoxdarā* в текстах).

В редких случаях тексты как будто предлагают глоссировать определённые глагольные формы некорректно с точки зрения словаря. Например, токен *deδ-d* в тексте разбирается как входить-PST, несмотря на то, что в списке корней для претерита в словаре нет корня *deδ-*, а есть только *de-*. Токен *x̌êy-d* отглоссирован как читать-3SG, но для формы 3 лица ед. ч. настоящего времени

в словаре предлагается только корень *x̌oy-*, а не *x̌ê̆y-* (*x̌ê̆y-* — корень для претерита). По-видимому, такие расхождения связаны с ошибками при глоссировании текстов или ошибками в словаре.

Некоторые явления шугнанской морфонологии оказались непокрыты анализатором в текущем его виде. Помимо распространённой сандхи [j] очень редко встречается также сандхи [w] (видимо, только рядом с огубленной гласной /u/). Токен *si-w-u* (три-SANDHI-CONJ), таким образом, не распознаётся. Возможно, неизвестные анализатору морфонологические явления скрыты и в токене *lův-ǰ-it* (говорить-PF-ADD), где вместо привычной клитики *-at* (ADD) используется *-it*. С другой стороны, возможно, это также ошибка записи или глоссирования текста.

Анализатор не учитывает двойственную природу некоторых клитик, которые внесены в список морфем, но могут также быть самостоятельными. Например, из-за этого верного разбора не получил токен *ik-am* (EMPH-1SG), где проклитика *ik-* используется в качестве корня, к которому присоединяется личная энклитика. Не учтены и другие редкие словообразовательные модели, в частности, композитные глаголы и имена (за исключением тех, что встречаются в словаре в качестве одного корня). По этой причине не распознались, например, токены *na-var-δe-d* (NEG-мочь-взять-3SG) с двумя глагольными корнями или *kata-x̌āb* (большой-ночь) с двумя именными. Особая словообразовательная модель обнаруживается и у числительных. Например, анализатор не предложил разборов для токена *nusad-u* (девятьсот-CONJ), который также можно разобрать как *nu-sad-u* (девять-сто-CONJ). В словаре нет корня *nusad*, а модель словосложения числительных анализатору пока неизвестна.

Наконец, большую проблему составляет графическая вариативность некоторых лексем. Из-за неё не были распознаны, например, токены *miloim* (*miloyim* ‘мягкий’ в словаре), *salomati* (*salůmati* ‘здоровье’), *tavallud* (*tawallud* ‘рождение’) и другие.

6. Заключение

В результате работы создан морфологический анализатор для шугнанского языка, доступный онлайн. Он работает на основе электронной версии словаря

Д. Карамшоева с привлечением дополнительных материалов (скрипта для преобразования словаря, словаря морфем). Анализатор выдаёт результат в нескольких форматах: в более удобном для пользователя текстовом формате и в виде файла .json, совместимого с корпусом шугнанского языка. Помимо анализатора, также доступен конвертер орфографий, который помогает в унификации текстов и решении некоторых проблем несоответствия графики.

Оценка результатов показала, что анализатор справляется с поставленной задачей и глоссирует значительную часть текста, поданного на вход. Для 69% токенов текстов, на которых программа тестировалась, был предложен верный разбор. В среднем около 29% токенов получили единственный разбор, который оказался верным. Вместе с тем в работе были обнаружены недостатки, которые предстоит решить в будущем: он не учитывает некоторые редкие особенности языка, не всегда справляется с сильной графической вариативностью текстов, а его словарный запас ограничен размерами словаря Д. Карамшоева, хоть и впечатляющими, но не до конца покрывающими многообразие лексики современной шугнанской речи. Дальнейшая работа над компьютерными ресурсами для шугнанского языка может помочь в решении этих проблем.

Литература

Каландаров 2020 — Т. Каландаров. Памирские языки: между прошлым и будущим (на примере шугнанского языка) // Отдел культурного наследия и гуманитарных наук УЦА, доклад № 6. Бишкек, 2020.

Карамшоев 1988 — Д. Карамшоев. Шугнанско-русский словарь в трёх томах. Том 1. М.: Наука, 1988.

Муравьёва 1975 — И. А. Муравьёва. Непозиционные чередования фонем в шугнанском языке // Исследования по структурной и прикладной лингвистике. М., 1975. С. 124—135.

Плунгян 2000 — В. А. Плунгян. Общая морфология: Введение в проблематику. М.: УРСС, 2000.

Эдельман 1999 — Д. И. Эдельман. Шугнанский язык // Н. Рогова (ред.). Иранские языки. III. Восточноиранские языки. Языки мира. М.: Индрик, 1999. С. 225—241.

Edelman, Dodykhudoeva 2009 — J. Edelman, L. Dodykhudoeva. Shughni Language // Gernot Windfuhr (eds.). The Iranian Languages. Oxon: Routledge, 2009. P. 787—825.

Ethnologue — Shughni. Ethnologue. https://www.ethnologue.com/language/sgh.

Приложения

Таблица 7. Сравнение проектной орфографии и орфографии словаря Д. Карамшоева

Проектная	**Карамшоев**
A/a	А/а
Ā/ā	Ӣ/ӣ
E/e	Е/е
Ê/ê	Ӗ/ӗ
I/i	Ę/ę
	И/и
Ī/ī	Ӣ/ӣ
O/o	О/о
U/u	Ǫ/ǫ
	У/у
Ū/ū	Ӯ/ӯ
Ů/ů	Ӱ/ӱ
W/w	W/w
Y/y	Й/й
B/b	Б/б
P/p	П/п
V/v	В/в
F/f	Ф/ф
G/g	Г/г
K/k	К/к
D/d	Д/д

Проектная	**Карамшоев**
T/t	Т/т
Δ/δ	Δ/δ
Θ/θ	Θ/θ
S/s	С/с
Z/z	З/з
Ў/ў	Ў/ў
Ӽ/ӽ	Ӽ/ӽ
Q/q	Қ/қ
Ғ/ғ	Ғ/ғ
X/x	Х/х
Ʒ/ʒ	Ҙ/ҙ
C/c	Ц/ц
J/j	Ҷ/ҷ
Č/č	Ч/ч
Ž/ž	Ж/ж
Š/š	Ш/ш
H/h	Ҳ/ҳ
L/l	Л/л
M/m	М/м
N/n	Н/н
R/r	Р/р

Изображение 1. Скриншот веб-интерфейса сайта «Компьютерные ресурсы для изучения шугнанского языка»