Записки научных
семинаров ПОМИ
Том 540, 2024 г.

А. Алексеев, А. Тиллабаева, Г. Дж. Кабаева, С. И. Николенко

ПРИМЕНЕНИЕ СИНТАКСИЧЕСКИХ ПАРСЕРОВ ДЛЯ ТУРЕЦКОГО ЯЗЫКА В ЗАДАЧЕ РАЗМЕТКИ КЫРГЫЗСКИХ СИНТАКСИЧЕСКИХ КОРПУСОВ

§1. Введение

Кыргызский (киргизский) язык, как и многие другие языки с малым количеством языковых ресурсов (less-resourced languages, low-resource languages; LRL) [6, 23, 38], в последнее время становится объектом внимания исследовательских коллективов, стремящихся улучшить машиночитаемые ресурсы и инструменты для его исследования. Одна из значительных проблем в этой области — разметка синтаксических корпусов (трибанков, treebanks), требующая существенных трудозатрат и времени экспертов.

В настоящей работе в качестве одного из вариантов решения этой проблемы мы предлагаем полуавтоматический подход к синтаксической разметке, основанный на переносе синтаксической разметки посредством перевода трибанка. Используя предварительную разметку, порождённую моделями, лингвисты могут сосредоточиться на её исправлении и доработке, что ускоряет процесс аннотирования в целом. Этот подход имеет потенциал значительно сократить усилия, необходимые для создания качественных синтаксических корпусов достаточно большого размера.

Мы демонстрируем эффективность этого подхода и предоставляем специализированный инструмент для переноса синтаксической разметки с более богатого ресурсами языка источника (турецкого языка) на целевой язык (кыргызский). Общие грамматические характеристики выбранных языков, в частности порядок слов в предложении и

Ключевые слова: грамматика зависимостей, обработка естественного языка, языки с малым количеством ресурсов, машинный перевод, обработка кыргызского языка.

Работа была поддержана грантом Российского Научного Фонда #22-11-00135, «Исследование и разработка технологий обработки и анализа мультимодальных неструктурированных данных из различных источников и их применимости для решения экономических и социальных задач».

агглютинативный строй, упрощают задачу межъязыкового переноса синтаксической разметки.

Наконец, мы оцениваем предложенную систему синтаксической разметки на основе недавно опубликованного в рамках проекта Universal Dependencies (UD) трибанка для кыргызского языка. Наши результаты показывают, что данный подход не только способен ускорить процесс «ручной» разметки, но и позволяет получать лингвистически значимые результаты, подчеркивая его потенциал для более широкого применения в условиях недостаточных ресурсов.

§2. Релевантные исследованию работы

2.1. Кыргызский язык в Universal Dependencies. В последние годы были достигнуты значительные успехи в адаптации фреймворка *Universal Dependencies* (UD) [12] для кыргызского языка. Продолжаются работы по расширению синтаксически размеченных корпусов для кыргызского языка и формализации руководств для аннотаторов, чтобы решить такие задачи, как токенизация копулы, разметка модальных слов, клаузы с нулевой вершиной и разграничение между словоизменением и словообразованием. Эти вопросы аналогичны тем, которые возникают при разработке ресурсов UD для других тюркских языков, и стали предметом ряда недавних исследований [43–45]. Приведенные ресурсы и руководства сыграют ключевую роль в последующем обучении парсеров и развитии инструментов для синтаксического анализа кыргызского языка в целом.

На ноябрь 2024 года под эгидой фреймворка UD были разработаны два значимых кыргызских синтаксических корпуса:

UD_Kyrgyz-KTMU Treebank. Этот трибанк [8] в первой включавшей его версии UD содержал 781 предложение, на момент конца октября 2024 года — 2480 предложений, размеченных в рамках грамматики зависимостей [33]. Описание набора данных, который далее будем условно называть *KTMU*, представлено в работе [9].

Kyrgyz-TueCL Treebank. Этот трибанк содержит 145 предложений, включая 20 предложений из набора *Cairo* [24] и около 100 предложений, предоставленных *UD Turkic Group*[1]. Каждое предложение

[1]Дополнительную информацию можно найти на сайте `https://github.com/ud-turkic`.

сопровождается переводами на английский, турецкий и азербайджанский языки, что делает этот набор данных частью более широкой инициативы UD Turkic Treebank [4, 40].

Эти ресурсы представляют собой важные шаги на пути к созданию надежных синтаксических инструментов для кыргызского языка и его интеграции в более широкую экосистему Universal Dependencies. Однако подходы к разметке в них сильно различаются. Некоторые различия в разметке были перечислены в работе [44], в частности, упоминались различия в разметке копулы (словоформы «эле», «болгон»), модальных слов «да», «эле», «керек», «бар», «жок».

Кроме перечисленных случаев в датасете *TueCL* у глаголов выделяют в отдельный токен дискурсивную частицу «–бы» (например, «жатабы», «б-екен»), в то время как в *KTMU* этой вопросительной частице не приписываются дополнительные пометы.

В трибанке *TueCL* авторы анализируют отрицательное слово «эмес» как наречие (`ADV`) в роли наречного модификатора (`advmod`). А в корпусе *KTMU* автор присваивает всем перечисленным словам помету глагол (`VERB`). При этом синтаксическая роль частицы размечается неконсистентно и встречается со следующими пометами: `compound:svc`, `acl`, `ccomp`, `nmod`.

Также различается разметка послелогов (например, «үчүн», «чейин», «соң»). Если в *TueCL* эти слова получают помету `ADP` и синтаксическую роль `case`, то в корпусе *KTMU* слова «үчүн» и «чейин» рассматриваются как наречия в роли наречного модификатора (`advmod`), а «соң» — как существительное (`NOUN`) в роли именного модификатора (`nmod`).

Это не исчерпывающий список различий в разметке между датасетами. Приведенные выше примеры дают лишь общее представление о большом объёме несоответствий в трибанках кыргызского языка.

2.2. Перенос синтаксической разметки. Большой обзор существующих методов трансфера синтаксической разметки был проделан в работе [11]. В нём авторы выделяют три основных подхода: перенос модели (model transfer), проекция аннотации и перевод трибанка. Первый подход подразумевает обучение моделей на данных языка источника, после чего полученная модель применяется для разметки целевого языка. В таком случае модель обучается на PoS-тегах (part-of-speech tags, частеречные метки), иногда с дополнением морфологических признаков слов.

В наиболее простом виде подход model transfer применяется в работе [32], где авторы обучают модель на предложениях, в которых все слова заменены соответствующими частеречными метками (PoS-тегами), предсказывать синтаксические роли. Затем модель применяют к предложениям целевого языка, а на последнем шаге обогащают полученные разборы лексикой.

Лучших результатов можно добиться добавлением лексической информации в модель при обучении, как с использованием дополнительных глосс для каждого слова [15, 42], так и с использованием многоязычных (мультилингвальных) векторных представлений [7].

Другим представляющим интерес подходом является проекция аннотаций, предполагающая наличие параллельного корпуса для выбранных языков. Предложения на исходном языке размечаются с помощью одноязычного (монолингвального) парсера, после чего синтаксическая разметка переносится на выравненные пословно предложения целевого языка [18]. Впоследствии обучается модель на полученных синтаксических деревьях целевого языка. Такой метод уже применялся для аннотирования предложений на кыргызском языке, взятых из корпуса эпоса «Манас» [35].

Проблемой при таком подходе является синтаксическая разница между языками (в упомянутой работе, в частности, между кыргызским и русским), вследствие которой задача пословного выравнивания предложений становится нетривиальной.

В работе [3] авторы показывают, что эту проблему можно решить, используя несколько языков в качестве источников и комбинируя многоязычные проекции, основанные на взвешенных оценках схожести языков-источников с целевым языком. В работе [26] совмещают методы проекции аннотаций и трансфера моделей, преобразуя предложения исходного языка в соответствии с преобладающими направлениями синтаксических зависимостей, полученными при проекции аннотаций на целевой язык, что позволяет преодолеть синтаксическую разницу между языками.

Наконец, третий подход — перевод трибанка — похож на метод проекции аннотаций, но для него параллельный корпус формируется с помощью машинного перевода [34], который может быть выполнен пословно или на уровне предложения. В статье в качестве предложений для тренировки модели берут золотой стандарт трибанка исходного

языка, после чего оценивают несколько вариантов пословного перевода этого трибанка на целевой язык. Авторы работы [22] указывают на важность наличия согласованных размеченных трибанков для межъязыкового переноса синтаксической разметки, а также составляют такой мультиязычный трибанк из пяти европейских и корейского языка.

Наиболее близким к целям нашей статьи является метод перевода трибанка, так как для кыргызского языка на данный момент доступен трибанк только очень ограниченного объема (см. раздел 2.1). При этом наша работа отличается от [34] выбором целевого языка из тюркской семьи, для языков которой доступно меньше трибанков по сравнению с европейскими языками. Кроме того, в настоящий момент работа по унификации разметки между языками этой группы ещё только начинается. Поэтому мы не ставим целью этой статьи обучение моноязычной модели, а останавливаемся на этапе создания инструмента для облегчения ручной разметки.

§3. Предлагаемый алгоритм

Описанный ниже метод представляет собой простую цепочку (пайплайн) для синтаксического разбора (в рамках формализма грамматики зависимостей) в условиях недостаточности ресурсов для обучения; для этого предлагается использовать модели, обученные на родственных и более богатых ресурсами языках. Этапы цепочки перечислены ниже.

(1) *Поиск набора данных для оценки качества:* определить подходящий датасет для оценки целевого языка.
(2) *Выбор модели для исходного языка:* выбрать предварительно обученную модель для синтаксического разбора в рамках того же формализма для синтаксически близкого языка.
(3) *Перевод целевого текста:* перевести текст на исходный язык (автоматически).
(4) *Перенос дерева разбора*: с помощью методов автоматического выравнивания параллельных текстов (bitext alignment) и простых алгоритмических преобразований перенести синтаксическую разметку на предложение на целевом языке.
(5) *Оценка построенного синтаксического дерева* с использованием стандартных метрик и скриптов для оценки качества полученного синтаксического дерева на основе исходного датасета;

это даст представление о числе необходимых правок, которые придётся выполнить аннотатору.

3.1. Выбор набора данных для оценки качества. В разделе 2.1 мы отметили наиболее значимые различия между доступными кыргызскими трибанками, и, на наш взгляд, наиболее подробна и непротиворечива разметка в трибанке *TueCL*. Кроме того, в качестве одной из базовых моделей мы использовали единственную доступную кыргызскую модель Stanza [25] `ktmu-nocharlm`, обученную на корпусе *KTMU*, поэтому его использование для оценки качества было бы некорректным.

3.2. Синтаксические парсеры для турецкого. При выборе парсеров мы ориентировались прежде всего на свежесть и размер корпуса для обучения и простоту использования соответствующего инструмента. Из основных доступных инструментов для синтаксического анализа, работающих с форматом и формализмом Universal Dependencies, мы выбрали парсеры библиотеки *Stanza* [25], обученные на трибанках из версии UD 2.12 *BOUN* [36] и *IMST* [29–31], так как эти наборы данных, на наш взгляд, ближе прочих по схеме разметки к *TueCL*. Также были апробированы модели UDPipe, обученные на версии UD 2.5 (также трибанк *IMST*).

3.3. Система перевода. В качестве систем перевода мы рассматривали наиболее популярные сервисы, предоставляющие услуги машинного перевода — Google.Translate и Яндекс.Перевод[2]. Также мы использовали перевод с помощью GPT4o [2]; соответствующий запрос (prompt), в котором требуется сохранять порядок и количество слов, представлен в таблице 1. Не все полученные таким образом переводы удовлетворяли требованиям, однако, судя по результатам, представленным в разделе 4, описанный приём оказал существенное влияние на качество итогового синтаксического переноса.

[2]От последнего пришлось отказаться, так как при переводе предложений было множество откровенно неверных переводов (к примеру, некоторые имена заменялись на не связанные по смыслу существительные); отметим, однако, что перевод на кыргызский и с кыргызского в Яндекс.Переводе ещё в бета-версии.

```
Here are the sentences in Kyrgyz.

Кыз досуна кат жазды.
Жамгыр жаап жатат окшойт.

<...>

Дениз уктатылды.
Алар кетти.
Ал кетти.

I want you to translate them to Turkish
line-by-line, but you must not change
the word order and the total number
of words in each sentence. Do not add
any extra comments.
```

Таблица 1. Запрос к ChatGPT4o с целью получения переводов предложений на кыргызском языке из трибанка *TueCL* на турецкий язык с сохранением порядка и числа слов в предложении.

3.4. Перенос синтаксической структуры. Для полноценного заполнения всех принятых в проекте Universal Dependencies полей для каждого слова в разборе нужно также получить лемму каждого слова. Поэтому, помимо переноса разметки, описанного выше, нами была подготовлена лемматизация с помощью морфологического анализатора apertium-kir [41]. Из предлагаемых инструментом разборов для простоты выбирался первый. Для предложенных далее эвристик также потребуется метод определения части речи (необязательно точный); для этого в качестве приближения PoS-теггера был также использован apertium-kir.

К этому этапу тексты должны быть токенизированы, а турецкие переводы — получены из *TueCL*, GPT-4o и Google Translate. Эти переводы должны быть обогащены синтаксической разметкой в формате UD с помощью моделей Stanza-IMST-charlm, Stanza-IMST-BERT, Stanza-BOUN-BERT, а также UDPipe-1 (см. раздел 3.2).

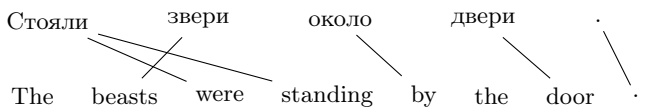

Рис. 1. Пример выравнивания предложений на русском и английском языках.

Далее следует произвести пословное выравнивание [10] между предложениями на турецком и кыргызском языках и преобразовать его эвристическими алгоритмами для последующего переноса синтаксических меток и структур с турецкого источника на целевые предложения на кыргызском языке. Подробное описание этого процесса дано далее.

3.4.1. *Выравнивание токенов на разных языках.* Мы применяли выравнивание (bitext alignment) с использованием SimAlign [19], чтобы сопоставить слова на кыргызском языке с турецкими и наоборот. Результат такой процедуры в общем случае является отношением «многие ко многим» (many-to-many), то есть произвольным двудольным графом, где рёбра проведены между парами слов из разных языков. Следует отметить, что мы могли бы использовать более современную модель, например, [14], и для общего улучшения качества сопоставления токенов дообучить её на парах предложений (например, построив с помощью машинного перевода параллельный турецко-кыргызский корпус), однако мы предпочли интерфейс с минимальными дополнительными настройками. Для выполнения выравнивания в качестве базовых моделей применялись многоязычная модель RoBERTa (XLM-R) [20,21] и mBERT (многоязычная модель на основе BERT) [13]. Пример выравнивания для предложений на русском и английском языках представлен на рисунке 1; обратите внимание, что в этом примере некоторым словам (артиклям “the”) невозможно сопоставить слово в другом языке, а некоторым словам приходится сопоставить несколько слов одновременно («стояли» и “were standing”). Также заметно, насколько «свобода» порядка слов и, в целом, разница в допустимом синтаксическом строе предложений в русском и английском языках влияет на степень «запутанности» выравнивания. В том числе поэтому в качестве «вспомогательного» языка для синтаксического разбора кыргызских предложений был выбран турецкий.

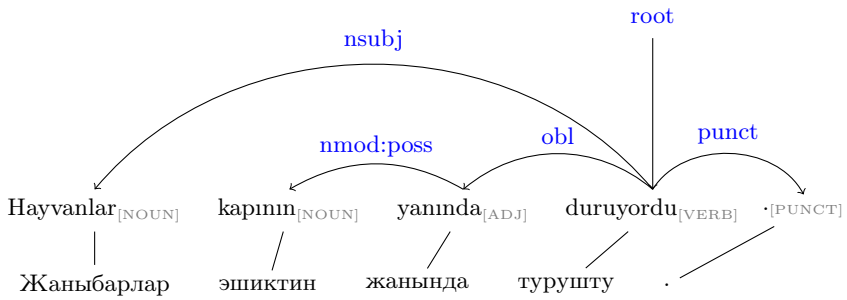

Рис. 2. Переводы предложения «Стояли звери около двери» на турецкий и кыргызский языки; порядок слов в этих языках очень схож. Над предложением на турецком показан его разбор в Universal Dependencies, полученный с помощью UDPipe [28] (модель `turkish-imst-ud-2.12-230717`, обученная на данных трибанка UD-IMST [29, 31]).

Таким образом, на данном этапе мы получаем приблизительное выравнивание между предложениями на турецком и кыргызском языках, а также деревья зависимостей, где вершинами являются токены из турецких предложений.

В идеальном случае, при совпадении количества токенов и выравнивании в виде тождественной перестановки, результат будет выглядеть как на рис. 2. Однако в большинстве случаев это не так: выравнивания содержат множество рёбер между словами на несоответствующих позициях, сопоставлений одному токенов нескольких, как на рис. 1 и 3, а также некоторое количество ошибочных сопоставлений. Поэтому мы разработали несколько правил переноса разметки, основанных на практических соображениях и не подстраиваясь под конкретную выборку (*TueCL*), о чём рассказывается ниже.

Помимо собственно точности переноса дуг и меток, необходимо учитывать требования к структуре. Дерево должно быть ациклическим и иметь корректную вершину предложения (или корень предложения, токен `ROOT`), что является нетривиальной задачей, поскольку для турецких `ROOT`-токенов не всегда находится соответствие в кыргызских предложениях из-за ошибок выравнивания или, в целом, структурного

несоответствия. Эти требования удовлетворяются при помощи следующих эвристик, без донастройки на тестовой выборке.

3.4.2. *Выделение вершины предложения.* В автоматическом разборе находится вершина турецкого предложения и проверяется, сопоставлен ли ей какой-либо токен в кыргызском предложении.

(1) Если сопоставление одно, пара исключается из общего выравнивания, гарантируя её включение в финальное представление (на следующем этапе выполняется построение паросочетания, см. далее).
(2) Если турецкому токену ROOT не сопоставлено ни одного токена, для его «жадного» выбора и назначения выполняется обратный обход кыргызского предложения (учитывая структуру SOV), назначая приоритет следующим образом:
    - токен той же части речи;
    - глагол;
    - существительное;
    - первое слово в предложении.
(3) Если турецкому токену ROOT сопоставлено несколько токенов, мы выбираем тот, чей порядковый номер ближе к номеру токена ROOT в турецком предложении. Это гарантирует включение вершины турецкого предложения в итоговое сопоставление.

3.4.3. *Фильтрация выравниваний.* Для турецких токенов с несколькими сопоставлениями используется POS-аннотация из apertium-kir (конвертированная в Universal Tagset с помощью apertium2ud [5]), чтобы отфильтровать сопоставления, оставляя только те, что соответствуют частям речи. Если такая фильтрация обнуляет количество сопоставлений, сохраняется исходный набор.

3.4.4. *Построение паросочетания.* С помощью библиотеки SciPy [39] методом Хопкрофта-Карпа [17] автоматически строится паросочетание, которое в дальнейшем и используется в качестве выравнивания для переноса меток, чтобы избежать циклов в дереве зависимостей при переносе разметки. Так гарантируется корректность структуры, хоть это потенциально и может привести к потере информации.

3.4.5. *Корректировка вершины предложения (технический шаг).* Для кыргызского токена, соответствующего турецкому токену ROOT согласно построенному паросочетанию, вручную задаётся `head` = 0.

3.4.6. *Перенос разметки.* В разметку для соответствующих кыргызских токенов переносится вся информация из турецкого предложения, за исключением `id`, `Token` и `Lemma`. Поле `head` затем обновляется так, чтобы отражать дуги из синтаксического разбора турецкого предложения.

Предложенный подход, основанный на эвристических правилах, обеспечивает высокую степень универсальности. Подробнее возможности улучшения результата переноса путём настройки эвристик на отложенной выборке обсуждаются в разделе 6.

3.5. Оценка качества. К настоящему моменту существуют удобные средства разметки деревьев зависимостей [16, 27], в том числе ориентированные в первую очередь на Universal Dependencies [37]. Для оценки труда экспертов при переразметке трибанков, подготовленных предложенным нами методом, подходят стандартные оценки качества синтаксического разбора в рамках грамматики зависимостей.

Так, UAS — группа оценок доли верно проведённых дуг (рёбер) — показывает, какую долю дуг придётся удалять (точность) а какую долю добавить заново (полнота). LAS — аналогичная, но более строгая оценка, учитывающая метки типов зависимостей на рёбрах: какую долю придётся либо перепроверить, либо отмечать другим типом.

Аналогичным образом можно интерпретировать точность и полноту токенизации (создание и удаление узлов в дереве), точность и полноту предсказания частеречных помет (создание и удаление UPOS-тегов), а также иных помет.

Для оценки качества был использован трибанк *TueCL*, так как в настоящий момент выбранные экспертами принципы разметки отражают, на наш взгляд, наиболее полно особенности синтаксиса кыргызского языка, которые необходимо учитывать при разметке по крайней мере этих предложений.

Для воспроизводимости, исключения ошибок и, в целом, удобства в качестве средства оценки качества был использован скрипт `UniversalDependencies/tools/eval.py` [1].

МТ	Выр.	Парсер	UAS Pr	UAS Re	UAS F1	LAS Pr	LAS Re	LAS F1	UPOS Pr	UPOS Re	UPOS F1
Кыргызский язык											
—	—	$St_{KTMU,\ nclm}$	49.02	47.65	48.33	29.29	28.47	28.88	68.04	66.13	**67.07**
Турецкий язык											
GPT4o	SA_{BERT}	$St_{BOUN,\ BERT}$	57.30	55.64	56.46	45.99	44.66	45.31	61.93	60.14	61.02
		$St_{IMST,\ BERT}$	55.04	53.45	54.23	43.21	41.96	42.57	61.83	60.04	60.92
		$St_{IMST,\ charlm}$	51.54	50.05	50.79	39.61	38.46	39.03	61.83	60.04	60.92
		UDPipe	48.46	47.05	47.74	26.65	25.87	26.25	56.07	54.45	55.25
	SA_{XLMR}	$St_{BOUN,\ BERT}$	60.39	58.64	**59.50**	48.25	46.85	**47.54**	63.17	61.34	62.24
		$St_{IMST,\ BERT}$	57.82	56.14	56.97	45.68	44.36	45.01	63.07	61.24	62.14
		$St_{IMST,\ charlm}$	54.32	52.75	53.52	41.87	40.66	41.26	63.07	61.24	62.14
		UDPipe	50.62	49.15	49.87	27.37	26.57	26.96	57.10	55.44	56.26
Google Translate	SA_{BERT}	$St_{BOUN,\ BERT}$	55.04	53.45	54.23	41.05	39.86	40.45	57.10	55.44	56.26
		$St_{IMST,\ BERT}$	52.47	50.95	51.70	39.09	37.96	38.52	57.10	55.44	56.26
		$St_{IMST,\ charlm}$	50.21	48.75	49.47	35.80	34.77	35.28	57.10	55.44	56.26
		UDPipe	48.25	46.85	47.54	23.66	22.98	23.31	53.81	52.25	53.02
	SA_{XLMR}	$St_{BOUN,\ BERT}$	56.17	54.55	55.35	42.59	41.36	41.97	57.82	56.14	56.97
		$St_{IMST,\ BERT}$	53.19	51.65	52.41	40.12	38.96	39.53	57.82	56.14	56.97
		$St_{IMST,\ charlm}$	50.62	49.15	49.87	36.83	35.76	36.29	57.82	56.14	56.97
		UDPipe	47.74	46.35	47.03	22.84	22.18	22.50	54.01	52.45	53.22
Переводы из *TueCL*	SA_{BERT}	$St_{BOUN,\ BERT}$	51.65	50.15	50.89	40.53	39.36	39.94	58.64	56.94	57.78
		$St_{IMST,\ BERT}$	47.94	46.55	47.24	37.96	36.86	37.40	58.54	56.84	57.68
		$St_{IMST,\ charlm}$	43.21	41.96	42.57	33.64	32.67	33.15	58.54	56.84	57.68
		UDPipe	43.42	42.16	42.78	22.53	21.88	22.20	54.42	52.85	53.62
	SA_{XLMR}	$St_{BOUN,\ BERT}$	53.50	51.95	52.71	42.39	41.16	41.76	57.72	56.04	56.87
		$St_{IMST,\ BERT}$	50.72	49.25	49.97	39.71	38.56	39.13	57.61	55.94	56.77
		$St_{IMST,\ charlm}$	45.68	44.36	45.01	34.98	33.97	34.47	57.61	55.94	56.77
		UDPipe	45.47	44.16	44.80	23.66	22.98	23.31	53.70	52.15	52.91

Таблица 2. Оценки качества: **Pr** — точность, **Re** — полнота, **F1** — F-мера; SA — SimAlign; UDPipe — парсер UDPipe-1, модель `IMST-UD-2.5-191206`, $St_{[treebank],\ [model]}$ — модели Stanza.

§4. Результаты

Все эксперименты были проведены на компьютере с 16 GB RAM, Intel i7-8565U CPU @ 1.80GHz с 4 ядрами и 8 лог. процессорами. Результаты экспериментов представлены в таблицах 2 и 3. В таблице 3 показаны метрики качества, для которых результаты всех моделей с переносом синтаксической разметки совпадают, так как этим метрики оценивают элементы разбора, полученные с помощью apertium-kir — для всех подходов, кроме Stanza `ktmu-noncharlm` (то есть единственного рассмотренного нами парсера непосредственно для кыргызского языка). Метрика "Слова" оценивает совпадение выравненных слов, а метрика "Леммы" оценивает, верно ли слова были приведены к словарной форме.

Поскольку предложенная цепочка обработки текстов состоит из нескольких шагов, не представляется возможным достаточно достоверно оценить степень влияния ошибок на каждой из них на итоговый результат. На достоверность выводов, вероятно, влияет также и небольшой размер тестовой выборки. Однако полученные результаты позволяют сделать следующие выводы:

- для данной задачи модель XLM-RoBERTa демонстрирует лучшее качество выравнивания в режиме zero-shot по сравнению с mBERT; к примеру, для F1-меры в задаче предсказания UPOS выравнивание с помощью XLM-RoBERTa дало преимущество (не более 1.22%) в 8 из 12 случаев (это не так лишь для турецких предложений из *TueCL*), в оценках UAS — в 10 из 12 случаев, в оценках LAS — в 11 из 12;
- для задач синтаксического анализа модели, использующие скрытые представления BERT, повсеместно показали лучшие результаты среди протестированных подходов;
- определение частей речи по итогам синтаксического переноса ожидаемо оказалось недостаточно эффективным (впрочем, оно и не было главной целью данной работы), и в этой задаче предпочтение следует отдавать apertium-kir или разметке с помощью Stanza-KTMU-nocharlm; предсказание частей речи (меток UPOS) средствами Stanza-KTMU-nocharlm позволяет достигуть точности 68.04%, полноты 66.13% и F_1-меры 67.07%, в то время как лучшие результаты, полученные с помощью переноса разметки моделью Stanza-BOUN-BERT, равны соответственно $\mathbf{Pr} = 63.17\%$, $\mathbf{Re} = 61.34\%$ и $\mathbf{F}_1 = 62.24\%$;

		Леммы			Слова		
	Модель	**Pr**	**Re**	**F1**	**Pr**	**Re**	**F1**
KY	$\text{St}_{\text{KTMU, nclm}}$	74.61	72.53	73.56	96.81	94.11	95.44
TR	Other (apertium-kir)	75.82	73.63	74.71	97.02	94.21	95.59

Таблица 3. Метрики качества: **Слова** — совпадение выравненных слов, **Леммы** — верно ли слова были приведены к словарной форме.

- использование машинного перевода с особой инструкцией для GPT4o дало заметный прирост качества; это также легко видеть по повсеместному преимуществу в значении F_1-меры для UAS, LAS и UPOS, перенос же с предложений на турецком языке почти повсеместно проигрывает, чего, в целом, следовало ожидать, так как при немашинном переводе нет факторов, потенциально полезных для сохранения сходного порядка слов (пусть и ценой точности перевода и грамматической корректности).

Особенно примечательным результатом стало превосходство в предсказании устройства деревьев зависимостей, полученных в рамках нашего подхода, над деревьями, сформированными кыргызским парсером Stanza-KTMU-nocharlm. Этот результат вдохновляет на дальнейшие исследования и разработку более совершенных парсеров.

Кроме прочего, полученные результаты открывают новые возможности для дальнейшего улучшения моделей синтаксического анализа и их адаптации под языки с ограниченными ресурсами.

§5. Анализ ошибок

В настоящем исследовании не предусматривается дополнительных эвристик для токенизации многословных выражений на отдельные токены, из-за чего 19.3% предложений были неверно токенизированы в сравнении с «золотым стандартом». Эти предложения не учитывались в последующем анализе ошибок. Анализируемые предложения на турецком, с которых проецировались аннотации на кыргызский, разобраны моделью Stanza-IMST, продемонстрировавшей высокое качество в эксперименте.

В таблице 4 приводится точность переноса типов синтаксических отношений (`deprel`) и точность определения для них вершин (`head`). При подсчете верных тегов `deprel`, кроме точности универсальных помет, мы учитывали также точность указания подтипа синтаксического отношения при его наличии, что отличается от подхода к вычислению LAS и UAS пакетом UD tools, используемом в предыдущем разделе. Более строгий подход позволяет детальнее оценить качество переноса аннотаций, что полезно в контексте применения нашего метода к полуавтоматической разметке. Из таблицы видно, что чуть больше половины тегов `deprel` ни разу не были размечены верно. Важно отметить, что авторы трибанка ставили целью продемонстрировать грамматические особенности кыргызского языка, и в результате трибанк содержит сравнительно большое количество редких и потому сложных для автоматической разметки случаев. К тому же в трибанке есть типы отношений, специфичные для кыргызского языка; к ним относятся: `nsubj:pass`, `nsubj:outer`, `obl:cau`, `obl:tmod`, `compound:svc`. Все эти подтипы синтаксических связей (кроме `nsubj:outer`) не используются в трибанке IMST. Поэтому при разметке турецкая модель присваивала универсальную помету, не указывая подтип (например, `obl` вместо `obl:cau`), что при оценивании считалось за ошибку.

Кроме того, значительный процент ошибок появился из-за синтаксической разницы в переводах предложений. Так, 35,9% ошибок в определении тега `deprel` объясняются тем, что слову из кыргызского языка не удалось найти в пару соответствующее слово в турецком переводе, из-за чего токен в кыргызском оставался без пометы.

Таким образом, например, для маркера аналитической формы глагола (`aux`), несмотря на его частотность в трибанке, ни разу не удалось выбрать верную помету (см. табл. 4), причём в 71% из общего числа ошибок для этого тега причина заключалась именно в синтаксических различиях между языками: при переводе на турецкий язык использовалась синтетическая форма глагола, в то время как в предложениях на кыргызском — аналитическая, и соответственно число токенов в предложениях различалось. Такой пример показан на рис. 3: взаимное местоимение «бири-бирин» («друг друга») переводится на турецкий одним словом «birbirlerini», «андан соң» (that-ABL after) единственным словом «sonra», а прошедшее время глагола выражено аналитически — «чыгып кетишти» (ср. в турецком «çıktılar»). Кроме

	Всего тегов	**Верных deprel**	**Верных head**
punct	152	91%	57%
nsubj	118	67%	64%
root	117	76%	76%
obl	65	58%	62%
aux	56	0%	52%
obj	49	67%	67%
advmod	26	35%	50%
advcl	23	57%	52%
conj	19	53%	42%
nmod	17	18%	71%
nmod:poss	15	73%	87%
ccomp	14	0%	14%
case	12	25%	25%
amod	11	73%	45%
det	11	55%	82%
cc	10	90%	40%
advmod:emph	10	40%	50%
acl	10	20%	10%
xcomp	10	0%	70%
compound	10	0%	10%
nummod	7	100%	100%
csubj	7	0%	43%
fixed	6	0%	0%
obl:tmod	6	0%	100%
cop	6	0%	83%
parataxis	6	0%	50%
orphan	5	0%	40%
flat	2	100%	100%
obl:cau	2	0%	100%
mark	2	0%	100%
compound:lvc	2	0%	0%
nsubj:outer	2	0%	100%
discourse	2	0%	50%
compound:svc	2	0%	50%
nsubj:pass	2	0%	100%
vocative	1	0%	100%
appos	1	0%	0%
acl:relcl	1	0%	100%

Таблица 4. Доля корректных переносов аннотаций синтаксических отношений (`deprel`) и выбора вершины для токенов (`head`) в сравнении разметки *TueCL* и одной из трёх лучших моделей $\text{Stanza}_{\text{IMST, charlm}}$.

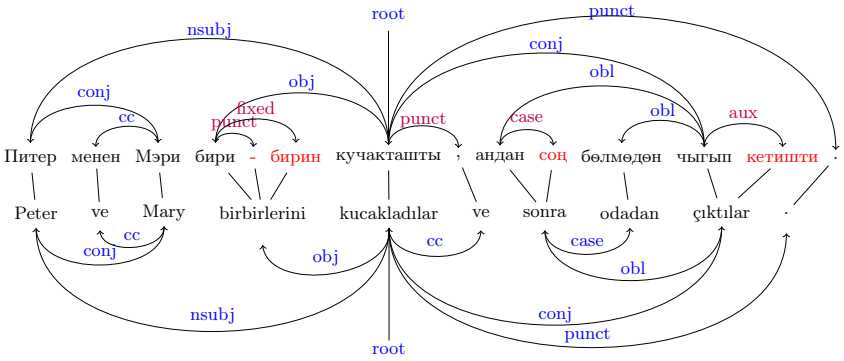

Рис. 3. Пример из трибанка *TueCL* (сверху) и его перевод на турецкий язык (снизу), размеченный моделью Stanza-IMST-charlm; красным выделены слова, которым не нашлось пары в турецком предложении, тёмно-красным — синтаксические отношения (`deprel`), неверно определённые при переносе разметки.

того, неправильный тег получает токен «запятая», поскольку в турецком этот токен «выравнен» с соединительным союзом «ve», стоящем на соответствующем месте. По этой же причине плохо определяются пометы, маркирующие вершину клаузы (`ccomp`, `xcomp`, `csubj`). При переводе на турецкий зачастую такие конструкции заменяются неклаузальными аналогами.

Также ошибки могут «накапливаться» и на стороне парсера турецкого языка, так, например, все отношения типа `parataxis` были размечены отношением `conj`.

Всего 10% ошибок в определении вершины токена были связаны с неправильным выравниванием предложений. Это значение существенно ниже в сравнении с аналогичным для `deprel` (35.9%), поскольку при невозможности определения пары для кыргызского слова в турецком предложении мы привязывали токен к вершине предложения (`ROOT`), а не оставляли это поле пустым, как в случае `deprel`.

§6. Заключение

6.1. Итоги работы. Настоящее исследование наглядно показывает, что машинный перевод в сочетании с моделями синтаксического парсинга, обученными на родственных языках, может существенно ускорить ручную разметку зависимостей для кыргызского языка. Сравнение подходов к пословному выравниванию, синтаксическому разбору и машинному переводу демонстрирует преимущество применения отдельных моделей в аналогичных решениях. Для выравнивания с помощью многоязычных векторных представлений без дообучения на парах слов лучше всего из рассмотренных подошла модель XLM-RoBERTa. Для синтаксического разбора турецких текстов из рассмотренных нами парсеров наиболее эффективен Stanza-BOUN-BERT. Для машинного перевода — ChatGPT4o с особой инструкцией (prompt), требующей у порождающей модели сохранять при переводе с кыргызского на турецкий порядок и число слов в предложении.

Учитывая острую необходимость в создании трибанков для кыргызского языка, пригодных для обучения парсеров, мы призываем исследователей и практиков изучать и развивать этот подход.

6.2. Ограничения. При оценке применимости предложенного метода следует учитывать следующие возможные ограничения представленного анализа.

Во-первых, предложенный нами подход не включает в себя дополнительных эвристик, адаптирующих проекции аннотаций специальным образом для кыргызского языка. Возможным улучшением этого инструмента было бы использование дополнительных правил, упрощающих разметку для ряда слов в кыргызском: дискурсивных маркеров, копулы, частиц, союзов, послелогов и т.д. Кроме того, специального способа обработки (в том числе написания правил токенизации, аннотации) требуют многословные выражения (multiword expressions), регулярно встречающиеся в кыргызском языке.

В будущем задачу токенизации многословных выражений можно решить с помощью инструмента apertium-kir [41], предоставляющего морфологические разборы слов в предложениях. Разрабатывать такие эвристики можно на небольшом наборе из восьми предложений[3],

[3]Размеченные предложения можно найти по ссылке `https://github.com/ud-turkic/general/blob/main/Annotations/Kyrgyz_JNW.conllu`.

не вошедших в корпус кыргызского языка *TueCL*, но также рассмотренных и аннотированных в рамках конференции *UD Turkic Group* в соответствии с правилами трибанка.

Во-вторых, данный метод, несмотря на свою простоту, подразумевает наличие разработанного морфологического анализатора, что в случае с по-настоящему малоресурсными языками не всегда представляется возможным. То же касается и использования ChatGPT для перевода предложений: для некоторых малоресурсных языков качество перевода всё еще остается невысоким, что может сильно повлиять на конечный результат.

В-третьих, в настоящей работе мы обозреваем лишь небольшое количество доступных моделей для выравнивания слов и синтаксических парсеров. Также открытым остается вопрос степени влияния исходного языка на полученную синтаксическую разметку, поэтому эксперименты с другими грамматически близкими кыргызскому языками ещё предстоит провести в будущих исследованиях.

На каждом этапе предложенной цепочки обработки существует значительное количество возможностей для накопления ошибок. В будущем следует проводить оценку качества на каждом из шагов или, по меньшей мере, исследовать степень влияния каждой из предложенных эвристик переноса аннотаций на итоговый результат (ablation study). Однако в рамках данной работы основное внимание уделялось демонстрации жизнеспособности предложенного подхода (*proof-of-concept*). Важно подчеркнуть, что любые накопленные ошибки приводят к пессимистичной оценке результатов, что предпочтительнее по сравнению с ситуацией, в которой результаты могли бы быть искусственно завышены.

Наконец, корпус *TueCL* слишком мал для убедительной оценки качества, а предложения в нём демонстрируют значительное сходство друг с другом, отличаясь лишь незначительными модификациями, которые иллюстрируют различные синтаксические явления. Поэтому в будущем было бы желательно использовать более репрезентативный трибанк, включающий длинные предложения и обладающий столь же тщательной разметкой. Идеальный корпус мог бы охватывать тексты различных жанров: художественная литература, научно-популярные тексты, новости, энциклопедии, социальные сети, поэзия и эпос. Однако в условиях ограниченных ресурсов мы вынуждены работать с доступными данными.

6.3. Идеи для дальнейших исследований. Предложенное эмпирическое исследование служит всего лишь обоснованием (proof of concept) подхода, и ясно, что это всего лишь первые шаги для успешного переноса синтаксической разметки. В будущем могут быть исследованы иные улучшения и альтернативные стратегии, например приведённые ниже.

Аннотирование с использованием LLM: прямое аннотирование с помощью подсказок для больших языковых моделей (LLM) представляет собой ещё одну перспективную возможность. Хотя неопубликованные результаты связанного исследования [35] показывают, что существующие генеративные модели часто дают сбои при попытках решать задачу методом zero-shot-learning («промптинг») и few-shot-learning (в частности, «промптинг с примерами»), мы уверены, что более детальное рассмотрение этого подхода ещё может раскрыть его потенциал для синтаксической разметки.

Набор автоматически построенных деревьев зависимостей как «серебряный стандарт»: описанные методы также могут быть использованы для создания «серебряного»[4] набора деревьев зависимостей, который можно будет использовать для обучения синтаксических парсеров кыргызского языка. Например, можно перевести турецкий корпус на кыргызский язык, выполнить выравнивание слов (bitext alignment), методом, аналогичным предложенному, перенести разметку на кыргызский перевод и использовать полученный набор данных для обучения парсера. Такой парсер, возможно, не достигнет идеальной точности, но он, вероятно, сможет добиться значительного улучшения в качестве по сравнению с текущим положением вещей (т.е. полным отсутствием автоматического синтаксического разбора или моделей, обученных на малом трибанке [8]).

Предложенные подходы открывают перспективные пути для решения критической нехватки синтаксических ресурсов для кыргызского языка. Развивая данный подход, дальнейшие исследования могут раскрыть более эффективные и точные методы создания корпусов зависимостей и обучения парсеров, помогая исследованиям и применениям методов обработки кыргызского языка и других языков с ограниченными ресурсами.

[4]Silver standard annotation — разметка, выполняемая моделями машинного обучения без или с частичной верификацией экспертами.

Список литературы

1. *UniversalDependencies/tools/eval.py: UD Evaluation Script on GitHub.* Available: `https://github.com/UniversalDependencies/tools/blob/19c980e95ed0944dd5ecd262322403f8a77cee69/eval.py`, 2024.
2. J. Achiam, S. Adler, S. Agarwal, et al., *GPT-4 technical report*, arXiv preprint arXiv:2303.08774 (2023).
3. Z. Agić, A. Johannsen, B. Plank, et al., *Multilingual Projection for Parsing Truly Low-Resource Languages*, Transactions of the Association for Computational Linguistics, No. 4 (2016), pp. 301–312.
4. A. Furkan, B. Chontaeva, C. Cöltekin, et al., *Unifying the Annotations in Turkic Universal Dependencies Treebanks*, 2nd UniDive Workshop Theses (Online), 2024.
5. A. Alekseev, *alexeyev/apertium2ud: mapping tagsets*, 2023.
6. A. Alekseev, T. Turatali, *KyrgyzNLP: Challenges, Progress, and Future*, Proceedings of the 12th International Conference on Analysis of Images, Social Networks, and Texts (AIST 2024). In print. To appear in Lecture Notes in Computer Science (LNCS 15419), Springer, 2024.
7. W. Ammar, G. Mulcaire, M. Ballesteros, et al., *Many Languages, One Parser*, Transactions of the Association for Computational Linguistics, No. 4 (2016), pp. 431–444.
8. I. Benli, *UD_Kyrgyz-KTMU: UD for Kyrgyz.* Available: `https://github.com/UniversalDependencies/UD_Kyrgyz-KTMU/`, 2023.
9. I. Benli, B. Sharshembaev, *Dependency Parsing Based Treebank for Kyrgyz Language*, Ymer, No. 23(7) (2024), pp. 325–342.
10. P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, R.L. Mercer, *The mathematics of statistical machine translation: Parameter estimation*, Computational Linguistics **19**(2) (1993), pp. 263–311.
11. A. Das, S. Sarkar, *A Survey of the Model Transfer Approaches to Cross-Lingual Dependency Parsing*, ACM Transactions on Asian and Low-Resource Language Information Processing **19**(5) (2020), pp. 1–60.
12. M.-C. de Marneffe, C.D. Manning, J. Nivre, D. Zeman, *Universal Dependencies*, Computational Linguistics **47**(2) (2021), pp. 255–308.
13. J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2019), pp. 4171–4186.
14. Z.-Y. Dou, G. Neubig, *Word Alignment by Fine-tuning Embeddings on Parallel Corpora*, Conference of the European Chapter of the Association for Computational Linguistics (EACL), 2021.
15. G. Durrett, A. Pauls, D. Klein, *Syntactic transfer using a bilingual lexicon*, Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (2012), pp. 1–11.
16. J. Heinecke, *ConlluEditor: a fully graphical editor for Universal Dependencies treebank files*, Universal Dependencies Workshop, Paris (2019).
17. J.E. Hopcroft, R.M. Karp, *An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs*, SIAM Journal on Computing **2**(4) (1973), pp. 225–231.

18. R. Hwa, Ph. Resnik, A. Weinberg, C. Cabezas, O. Kolak, *Bootstrapping parsers via syntactic projection across parallel texts*, Natural Language Engineering **11**(3) (2005), pp. 311–325.
19. M.J. Sabet, Ph. Dufter, F. Yvon, H. Schütze, *SimAlign: High Quality Word Alignments without Parallel Training Data using Static and Contextualized Embeddings*, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings (2020), pp. 1627–1643.
20. A. Conneau, *Unsupervised cross-lingual representation learning at scale*, arXiv preprint arXiv:1911.02116 (2019).
21. Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Che, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, *RoBERTa: A Robustly Optimized BERT Pretraining Approach*, arXiv preprint arXiv:1907.11692 (2019).
22. R. McDonald, J. Nivre, Y. Quirmbach-Brundage, et al., *Universal Dependency Annotation for Multilingual Parsing*, Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (2013), pp. 92–97.
23. J. Mirzakhalov, A. Babu, A. Kunafin, A. Wahab, B. Moydinboyev, S. Ivanova, M. Uzokova, Sh. Pulatova, D. Ataman, J. Kreutzer, F. M. Tyers, O. Firat, J. Licato, S. Chellappan, *Evaluating Multiway Multilingual NMT in the Turkic Languages*, Proceedings of the Sixth Conference on Machine Translation (2021), pp. 518–530.
24. J. Nivre, *Towards a universal grammar for natural language processing*, International conference on intelligent text processing and computational linguistics, Springer (2015), pp. 3–16.
25. P. Qi, Y. Zhang, Y. Zhang, J. Bolton, C.D. Manning, *Stanza: A Python Natural Language Processing Toolkit for Many Human Languages*, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (2020).
26. M.S. Rasooli, M. Collins, *Density-Driven Cross-Lingual Transfer of Dependency Parsers*, Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal (2015), pp. 328–338.
27. P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, J. Tsujii, *brat: a Web-based Tool for NLP-Assisted Text Annotation*, Proceedings of the Demonstrations Session at EACL 2012, Avignon, France (2012).
28. M. Straka, *UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task*, Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Brussels, Belgium (2018), pp. 197–207.
29. U. Sulubacak, G. Eryiğit, *Implementing universal dependency, morphology, and multiword expression annotation standards for Turkish language processing*, Turkish Journal of Electrical Engineering and Computer Sciences **26**(3) (2018), pp. 1662–1672.
30. U. Sulubacak, G. Eryiğit, T. Pamay, *IMST: A revisited Turkish dependency treebank*, The 1st International Conference on Turkic Computational Linguistics, Ege University Press (2016), pp. 1–6.
31. U. Sulubacak, M. Gökırmak, F. M. Tyers, Ç. Çöltekin, J. Nivre, and G. Eryiğit, *Universal dependencies for Turkish*, Proceedings of COLING 2016, the 26th

International Conference on Computational Linguistics: Technical papers (2016), pp. 3444–3454.
32. A. Søgaard, *Data point selection for cross-language adaptation of dependency parsers*, The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers-Volume 2 (2011), pp. 682–686.
33. L. Tesnière, *Éléments de syntaxe structurale*, Paris: Klincksieck (1959).
34. J. Tiedemann, Ž. Agić, J. Nivre, *Treebank Translation for Cross-Lingual Parser Induction*, Proceedings of the Eighteenth Conference on Computational Natural Language Learning, Ann Arbor, Michigan (2014), pp. 130–140.
35. A. Tillabaeva, *Syntactic Transfer Based on the Polivariant Parallel Kyrgyz-Russian Corpus Manas*, Master's thesis, HSE University, Moscow, Russia (2024). Available: `https://www.hse.ru/en/ma/ling/students/diplomas/930858853`.
36. U. Türk, F. Atmaca, Ş. B. Özateş, G. Berk, S. T. Bedir, A. Köksal, B. Ö. Başaran, T. Güngör, A. Özgür, *Resources for Turkish Dependency Parsing: Introducing the BOUN Treebank and the BoAT Annotation Tool*, Language Resources and Evaluation **56**(1) (2022), pp. 259–307.
37. F. M. Tyers, M. Sheyanova, J .N. Washington, *UD Annotatrix: An Annotation Tool for Universal Dependencies*, Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories (TLT16) (2018), pp. 10–17.
38. Y. Veitsman, *Recent Advancements and Challenges of Turkic Central Asian Language Processing*, Available: `https://arxiv.org/abs/2407.05006`, (2024).
39. P. Virtanen, R. Gommers, T. E. Oliphant, et al., *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, Nature Methods **17** (2020), pp. 261–272.
40. J. Washington, Ç. Çöltekin, F. Akkurt, B. Chontaeva, S. Eslami, G. Jumalieva, A. Kasieva, A. Kuzgun, B. Marşan, Ch. Taguchi, *Strategies for the Annotation of Pronominalised Locatives in Turkic Universal Dependency Treebanks*, Proc. Joint Workshop on Multiword Expressions and Universal Dependencies (MWE-UD), LREC-COLING 2024 (2024), pp. 207–219.
41. J. N. Washington, M. Ipasov, F. M. Tyers, *A finite-state morphological transducer for Kyrgyz*, LREC (2012), pp. 934–940.
42. H. Zhao, Y. Song, Ch. Kit, G. Zhou, *Cross-language dependency parsing using a bilingual lexicon*, Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Suntec, Singapore (2009), pp. 55–63.
43. Г. К. Джумалиева, А. А. Касиева, С. Ж. Мусажанова, *Адаптация терминов веб-проекта «Универсальные зависимости» на кыргызский язык.* — Вестник КРСУ **23**(6) (2023), pp. 71–75.
44. А. А. Касиева, Г. К. Джумалиева, А. Томпсон, et al., *Проблемы кыргызской синтаксической аннотации в фреймворке Universal Dependencies*, Proceedings of TurkLang-2023, Бухара (2023), pp. 189–216.
45. С. Ж. Мусажанова, А. А. Касиева, Г. К. Джумалиева, *Синтаксическая аннотация кыргызского языка на основе новосозданного корпуса.* — Вестник Иссык-Кульского университета **54**(2) (2023), pp. 140–148.

Alekseev A., Tillabaeva A., Kabaeva G. Dzh., Nikolenko S. I. Syntax transfer to Kyrgyz Using the Treebank Translation Method.

Kyrgyz is a less-resourced language and requires significant effort to create high-quality syntax corpora (treebanks). In this work, we propose an approach that simplifies the development of a treebank for the Kyrgyz language. We present a tool for transferring syntactic annotations from the Turkish language to Kyrgyz based on the treebank translation method. We evaluate the efficiency of our approach using the TueCL treebank. Results show that our method provides higher quality of syntactic annotation compared to a monolingual model trained on the Kyrgyz KTMU treebank. Moreover, in this work we propose a method to evaluate the complexity of manual annotation for the resulting syntax trees, contributing to further optimization of the annotation process.

Поступило 15 ноября 2024 г.

Санкт-Петербургское отделение
Математического института им. В. А. Стеклова РАН,
191023, наб.р. Фонтанки, 27, Санкт-Петербург, Россия;
СПбГУ, Факультет МКН
199178, 14-ая линия ВО, 29, Санкт-Петербург, Россия;
КФУ, Хим. институт им. А.М. Бутлерова
420008, ул. Кремлёвская, 18, Казань, РТ, Россия;
КГТУ им. И. Раззакова
720044, пр. Ч. Айтматова, 66, Бишкек, Кыргызстан (Киргизия)
E-mail: anton.m.alexeyev@gmail.com

Независимая исследовательница,
Бишкек, Кыргызстан (Киргизия)
E-mail: alinatillabaeva42@gmail.com

КГТУ им. И. Раззакова,
720044, пр. Ч. Айтматова, 66, Бишкек, Кыргызстан (Киргизия)
E-mail: kabaevagd9@kstu.kg

Университет ИТМО, Санкт-Петербург, Россия;
Санкт-Петербургское отделение
Математического института им. В. А. Стеклова РАН,
191023, наб.р. Фонтанки, 27, Санкт-Петербург, Россия
E-mail: sergey@logic.pdmi.ras.ru