УДК 004.89

НЕЙРОСИМВОЛИЧЕСКИЙ ПОДХОД К ДОПОЛНЕННОЙ ГЕНЕРАЦИИ ТЕКСТА НА ОСНОВЕ АВТОМАТИЗИРОВАННОЙ ИНДУКЦИИ МОРФОТАКТИЧЕСКИХ ПРАВИЛ

М. В. Исангулов[1] [0009-0006-3244-0328], А. М. Елизаров[2] [0000-0003-2546-6897], А. Р. Кунафин[3] [0009-0006-0495-265X], А. Р. Гатиатуллин[4] [0000-0003-3063-8147], Н. А. Прокопьев[5] [0000-0003-0066-7465]

[1, 2]Казанский (Приволжский) федерального университет, г. Казань, Россия
[3]Независимый исследователь
[4, 5]Академия наук Республики Татарстан, *г. Казань, Россия*

[1]marathon.our@gmail.com, [2]amelizarov@gmail.com, [3]aigizk@gmail.com, [4]ayrat.gatiatullin@gmail.com, [5]nikolai.prokopyev@gmail.com

Аннотация

Представлен гибридный нейросимволический метод, который объединяет большую языковую модель (LLM) и конечный автомат (FST) для обеспечения морфологической корректности при генерации текста на агглютинативных языках.

Система автоматически извлекает правила из корпусных данных: для локальных примеров словоформ LLM формирует цепочки морфологического разбора, которые затем агрегируются и упорядочиваются в компактные описания правил морфотактики (LEXC) и выбора алломорфов (regex). На этапе генерации LLM и FST работают совместно: если токен не распознается автоматом, LLM извлекает из контекста пару «лемма + теги», а FST реализует корректную поверхностную форму. В качестве набора данных использован корпус художественной литературы (~1600 предложений). Для списка из 50 существительных извлечено 250 словоформ. По предложенному алгоритму LLM сгенерировала 110 контекстных regex-правил вместе с LEXC-морфотактикой, на основе чего был скомпилирован FST, распознавший 170/250 форм (~70%). В прикладном тесте машинного перевода на подкорпусе из 300 предложений интеграция данного FST в цикл LLM повысила качество с BLEU 16.14 / ChrF 45.13 до BLEU 25.71 / ChrF 50.87 без дообучения переводчика. Подход применим к иным частям речи и другим

агглютинативным и малоресурсным языкам, где он может быть использован для наполнения словарных и грамматических ресурсов.

***Ключевые слова**: нейросимволический подход, большая языковая модель, конечные автоматы, двухуровневая морфология, LEXC морфотактика, машинный перевод, агглютинативные языки, башкирский язык.*

ВВЕДЕНИЕ

Морфологически сложные агглютинативные языки до сих пор остаются сложной областью для современных больших языковых моделей (LLM). Один корень может реализовывать десятки поверхностных форм за счет суффиксальной агглютинации, гармонии гласных, чередований и контекстных модификаций морфем. В условиях дефицита данных и неточной подсловной токенизации (BPE, unigram LM) модели часто не выделяют морфологически осмысленные сегменты; редкие суффиксальные цепочки нередко встречаются в обучении единично. Дополнительно агглютинативные языки слабо представлены в мультиязычных корпусах, что затрудняет обобщение поверхностных форм. В результате LLM порождают несуществующие словоформы, смешивают алломорфы и нарушают порядок морфем, что является проявлением «морфологического разрыва», когда смысл сохраняется, а форма деградирует [1]. На богатых ресурсами языках нейронные модели словоизменения (seq2seq/ трансформеры) справляются значительно лучше [2], однако их устойчивость на малоресурсных и агглютинативных системах остается ограниченной [3], что мотивирует гибридные решения.

Формальная линия работ по конечным автоматам восходит к двухуровневой морфологии Коскенниеми и инструментам Xerox [4, 5]. Конечные автоматы (FST) позволяют строго задать морфотактику (например, LEXC) и детерминированно реализовывать контекстные преобразования поверхностных форм. Такие грамматики теоретически порождают все допустимые формы и исключают недопустимые.

Современные открытые стеки (HFST LexC/TwolC; foma) поддерживают тот же парадигматический подход, включая компиляцию двухуровневых правил и разрешение конфликтов с весами, что делает FST практичными для морфологии

богатых языков. Тем не менее ручная разработка LEXC/правил алломорфии трудоемка и плохо масштабируется на новые части речи и языки. Популярные словарно-ориентированные корректоры орфографии, такие как Hunspell, используют форматы .dic (список слов с флагами) и .aff (аффиксальные директивы) и удобны для правки или проверки, но слабо генерализуются за пределы заданного словаря и не решают контекстный выбор граммем, поскольку опираются на перечисление слов и аффикс-паттернов, а не на полноценную морфологическую грамматику. В экосистеме Apertium доступны морфологические анализаторы и генераторы и shallow-transfer MT; для башкирского языка существует проект apertium-bak (bakmorph), однако типичный рабочий процесс остается лексикографически нагруженным и требует ручной поддержки XML словарей/парадигм.

Имеющиеся «нейро-символьные» стыковки, как правило, решают смежные, но иные задачи. В SGNMT конечные автоматы/решетки подключаются как «predictors» в декодере, то есть ограничивают поиск и добавляют внешние оценки, не гарантируя морфологически корректную поверхность и не выполняя выбор алломорфа в контексте [6]. В коррекции и спелл-чеке FST обычно выступает внешним источником разрешенных форм (аффиксальные словари, грамматики), не будучи интегрированным в сам генеративный цикл [7]. Существуют и двухшаговые NMT-схемы с выходом в формате lemma+tag и последующей детерминистической генерацией поверхности (например, с использованием SMOR для немецкого): такие подходы уменьшают разреженность, но разметка делается вручную, а FST не «учится» из корпуса и не встраивается как онлайн-валидатор/генератор [8]. Наконец, «обратное» направление — бутстрап нейронных анализаторов по готовым FST — демонстрирует рост покрытия и точности по сравнению с исходными автоматами, но не решает задачу индукции самих правил/лексиконов из корпусных свидетельств и их совместной работы с LLM в момент генерации [9]. В результате в текущем дискурсе отсутствуют работы, где а) LLM генерирует переносимую LEXC/regex-грамматику из корпусных примеров, б) этот автомат включен в сам процесс генерации для коррекции словоформ и в) показано заметное улучшение качества вывода на целевых корпусах.

В настоящей работе представлено решение описанных проблем в форме гибридного нейросимволического подхода — комбинации использования LLM и FST.

1. Из небольшой корпусной выборки извлекаются морфотактические описания (LEXC) и компактные контекстные правила (regex) выбора алломорфов: локальные правила, сгенерированные LLM по представленным формам, агрегируются, унифицируются и упорядочиваются от специфичных к обобщенным. Важное отличие от словарных систем (Hunspell и др.) является то, что конкретный список лемм служит лишь источником индукции, а полученные правила обобщаются на леммы вне словаря (например, шаблоны +ACC:+НЫ с контекстными правилами фонологической реализации применимы к любому существительному из класса), что уменьшает ручные затраты и повышает переносимость.

2. На этапе генерации текста компоненты работают в одном контуре: если токен не распознан автоматом, LLM извлекает из контекста пару «лемма + теги», а FST детерминированно реализует корректную поверхностную форму, тем самым сочетаются контекстная уместность и морфологическая валидность.

Эмпирически показаны значимый уровень корректного распознания словоформ, учитываемых в правилах, и прирост качества генерации текста на примере машинного перевода без дообучения модели (BLEU+9.57, ChrF+5.74 на подкорпусе художественного текста), при том, что грамматика остается компактной и объяснимой. Подход может быть применен к другим частям речи и агглютинативным, малоресурсным языкам, он предлагает практический путь к ускоренному пополнению словарных и грамматических ресурсов.

ГЕНЕРАЦИЯ ПРАВИЛ И АВТОМАТИЗИРОВАННЫЙ ПАЙПЛАЙН

Цель этого раздела — последовательно показать, как из небольшого параллельного корпуса автоматически извлекаются и анализируются словоформы, как на их основе синтезируются правила LEXC и regex и компонуется двунаправленный FST, который затем можно и анализировать (surface → analysis), и генерировать (lemma + tags → surface). Описаны также процесс индукции правил морфотактики LEXC и правил выбора алломофов regex, сборка

трансдьюсера и итеративная процедура расширения покрытия (схема пайплайна показана на рис. 1).

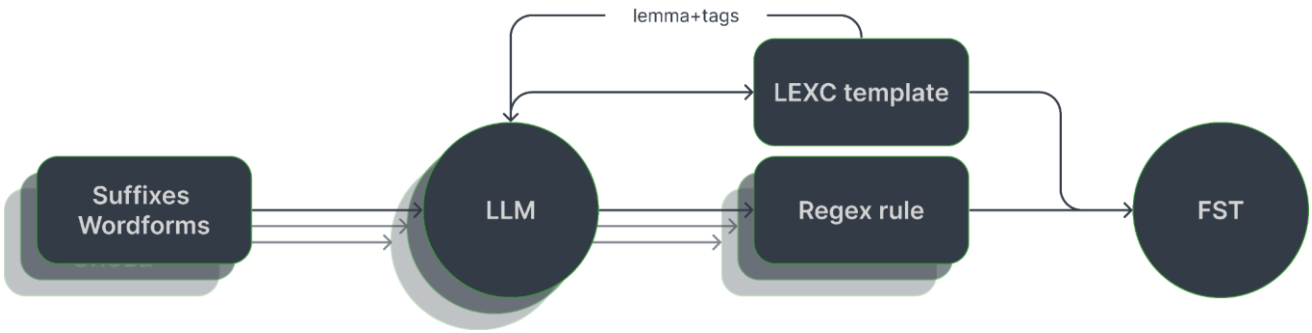

Рис. 1. Автоматизированная индукция: от словоформ к LEXC/regex и сборке FST.

Исходные данные — параллельное издание «Маленького принца» на башкирском и русском (~1600 предложений). Для иллюстрации приведен фрагмент:

Был китапты өлкән кешегә бағышлаған өсөн балаларҙан мине ғәфү итеүҙәрен һорайым. Прошу детей простить меня за то, что я посвятил эту книжку взрослому.

Для индукции был заранее отобран список целевых именных лемм:

бала кеше китап …

В настоящей работе сознательно фокус сделан только на существительных и описана их парадигма как последовательность состояний:

NStem → NNumber → NPoss → NCase → #

Корпус автоматически просматривается на предмет всех извлеченных словоформ для выбранных лемм в виде lemma, surface, позиция, локальный контекст. Пример структурированного вывода:

{"noun": "бала", "matches": [{"line_num": 4, "matched_form": "балаларҙан"},
{"line_num": 6, "matched_form": "балалар"}]}
{"noun": "йылға", "matches": [{"line_num": 865, "matched_form": "йылғалар"},
{"line_num": 882, "matched_form": "йылғаларға"}]}

Чтобы понять, какие алломорфы потребуются, были сгруппированы наблюдения по схожим суффиксальным хвостам и оценены их частоты. Это помогает выделить «опорные» аффиксы, для которых и будут записываться правила:

{"suffix": "лар", "count": 16, "occurrences": [{"form": "йылғалар", "noun": "йылға",
"line_num": 865}, {"form": "балалар", "noun": "бала", "line_num": 6}]}

На уровне морфотактики необходимы правила в формате LEXC, которые определяют допустимый порядок морфем. Такой шаблон сгенерирован с помощью LLM: верхний уровень задает путь NStem → NNumber → NPoss → NCase → #, а на surface-стороне вместо конкретных букв использованы шаблоны — абстрактные маркеры, которые позже «развернут» regex-правила. Например, +ЛАр означает множественное число, +ҺЫ — притяжательность с гармонией, +ГА/+ДА — классы онсета и гласной для дательного/местного, +НЫ — винительный с вариациями Н/ҙ/д/т и Ы/е по контексту. Эти шаблоны нужны, чтобы разделить ответственность: LEXC строго фиксирует порядок морфем, а выбор конкретных алломорфов определяется контекстным правилам (HFST/XFST regex), которые заменяют «заглавные» компоненты (Л, А, Ы, и т. п.) на нужные буквы в зависимости от окружения.

Минимальный фрагмент LEXC выглядит так:

```
LEXICON Root
бала   +N:0       NNumber ;
йылға  +N:0       NNumber ;
```

```
…
LEXICON NNumber
+SG:0      NPoss ;
+PL:+ЛАр    NPoss ;

LEXICON NPoss
+POSS0:0      NCase ;
+POSS1SG:+һЫ   NCase ;
+POSS2SG:+һЫ   NCase ;
+POSS3SG:+һЫ   NCase ;
+POSS1PL:+һЫ   NCase ;
+POSS2PL:+һЫ   NCase ;
+POSS3PL:+һЫ   NCase ;

LEXICON NCase
+NOM:0    # ;
+ACC:+НЫ  # ;
+GEN:+НЫ  # ;
+DAT:+ГА  # ;
+LOC:+ДА  # ;
+ABL:+ДА  # ;
```

Чтобы связать поверхности с анализами, каждой найденной форме нужна разметка «лемма + теги», которую размечает LLM, явно фиксируя формат и порядок тегов. Базовый системный промпт выглядит так:

```
"You label Bashkir NOUN analyses. Use this exact order:\n"
"Lexeme+N+{SG|PL}+{POSS1SG|POSS2SG|POSS3SG|POSS1PL|POSS2PL|POSS3PL|POSS0}?+{NOM|ACC|DAT|LOC|ABL|GEN}\n…"
```

В запрос также добавляется контекст из корпуса (оба языка):

```
prompt.append(f"- form: {it['form']} | noun: {it['noun']}  ba: {it['ba']}  ru: {it['ru']}")
```

На выходе получается:

```
балалар = "бала+N+PL+NOM"
баланы  = "бала+N+SG+ACC"
йылғаларға = "йылға+N+PL+DAT"
```

Далее по каждому «семейству» аффикса компактный пакет словоформ отправляется в LLM, чтобы сгенерировать правила, реализующие поверхностную алломорфию для соответствующего шаблона:

```
"You are designing HFST regex rules to realize noun surface allomorphy from
abstract LEXC tags …"
```

LLM выдает набор узконаправленных правил, которые затем объединяются и упорядочиваются «специфичное → основное». Примеры агрегированных правил для множественного числа и гармонии (схематично):

```
Л -> л || [ а | ә | ы | е | о | ө | я | э ] "+" _ [ A r ]
А -> а || [ а | о | у | ы ] ?* "+" [ л | т | д | ҙ ] _ [ r ]
```

Сборка трансдьюсера идет по цепочке: сначала компилируется LEXC, затем правила, после чего они последовательно композируются. В коде это отражается следующим образом:

```
# 1) компиляция морфотактики
lexc = hfst.compile_lexc_file(str(_built))
# 2) компиляция и композиция правил
# rules = ... compose_sequence(... hfst.regex(pattern) ...)
lexc.compose(rules)  # итоговый T = LEXC ∘ R1 ∘ R2 ∘ ... ∘ Rk
```

Получившийся FST на стороне анализа распознает данные словоформы и возвращает соответствующие разборы. После первой сборки автоматически вычленяются формы из корпуса, которые еще не распознаются, и повторяется цикл: LLM проставляет переходы, генерирует новые локальные правила, унифицирует, пересобирает. Итерации идут до тех пор, пока добавление правил

дает стабильный прирост покрытия, и размер автомата остается в заданных пределах.

На наборе из 50 лемм этот процесс привел к автомату, распознающему 170 из 250 уникальных словоформ (≈70%). В корпусе для этих лемм встретилось около 200 уникальных суффиксальных хвостов; итоговый стек правил включает порядка 110 regex-переписываний, покрывающих множественное (+ЛАр), притяжательность (1SG, 2SG, 3SG и 1PL, 2PL, 3PL) и падежи (ACC, GEN, DAT, LOC, ABL). Это обеспечивает переносимую LEXC/regex-грамматику, которая обобщается на леммы вне исходного списка и существенно автоматизирует построение конечных автоматов для именной морфологии.

ГИБРИДНАЯ АРХИТЕКТУРА ПЕРЕВОДА

Цель этого раздела — показать, как собранный в п. 2 двунаправленный FST практически используется вместе с LLM для автоматической правки морфологических ошибок при переводе. Двунаправленность важна: один и тот же автомат умеет анализировать поверхность (surface → lemma + tags) и генерировать корректную словоформу по анализу (lemma+tags → surface). Это позволяет не только проверять вывод модели, но и восстанавливать правильную форму там, где LLM дала «несуществующее» слово.

Формируется испытательный набор предложений из того же параллельного издания «Маленького принца» (~1600 пар предложений на русском и башкирском языках). Сначала извлекаем все предложения, содержащие извлеченные словоформы из целевой именной области, таких предложений набралось 300. Каждое из них переводится с русского на башкирский с помощью gpt-4o-mini, сохраняя исход и последующие правки для метрик. Базовая подсказка к переводу минимальна и нейтральна:

You are a professional translator. Translate the given Russian sentence into Bashkir.

Полученные гипотезы служат базовой линией качества; они же становятся входом для нашего гибридного цикла LLM ↔ FST. Основная идея такова: проверяется каждое сгенерированное с помощью FST слово; если автомата «нет» на эту поверхность, LLM возвращает «лемма + теги» для позиции, затем

генерируется корректная форма через FST и подставляется в перевод. Благодаря двунаправленности та же грамматика валидирует и исправляет (цикл показан на рис. 2).

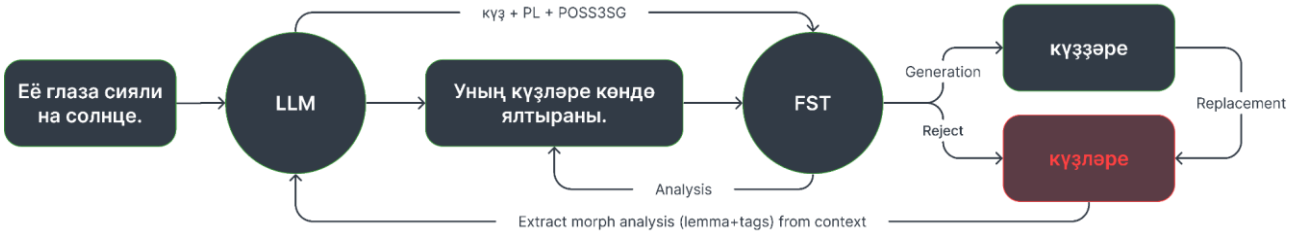

Рис. 2. Контур LLM ↔ FST при переводе: выявление нераспознанной формы, извлечение lemma+tags и генерация корректной поверхности.

Далее приведен пример работы цикла на одной фразе:

Запрос к переводчику:
Переведи с русского на башкирский: Её глаза сияли на солнце.

Исходный ответ gpt-4o-mini:
Уның күҙләре көндө ялтыраны.

Проблема:
күҙләре отсутствует в FST — форма несовместима с башкирской морфологией.

Уточнение к LLM (локальный контекст позиции):
Какая лемма и какие грамматические признаки у слова күҙләре в этом контексте?

Ответ:
күҙ + PL + POSS3SG

Генерация через FST:
күҙ + PL + POSS3SG → күҙҙәре

Итог:
Уның күҙҙәре көндө ялтыраны.

Смысл сохраняется, морфологическая ошибка снимается автоматически. Приведем еще один показательный случай с множественным числом:

До: Малайлар уҡытыусыга килде.
После: Малайҙар уҡытыусыға килде.

Тот же протокол формализуем простым псевдокодом:

```
for s in test_set:                    # 300 предложений
 hyp = LLM_translate(s.ru)            # gpt-4o-mini, базовый промпт
 for each token t at position p in hyp:
  if not FST.recognizes(t):
   (lemma, tags) = LLM_infer_analysis(s.ru, hyp, p)   # локальный контекст
   t_corr = FST.generate(lemma, tags)                 # lemma+tags → surface
   if t_corr exists:
    replace t with t_corr in hyp
 save {baseline: original LLM hyp, corrected: hyp}
```

Чтобы сравнить качество, для каждого сегмента сохраним базовую гипотезу и исправленную версию вместе с эталоном. Формат записи следующий:

{"line_num": 378, "translation": "Мин эште ташланым.", "fst_corrected": "Мин эшемде ташланым.", "reference": "Мин эшемде ташланым."}

На этом подкорпусе из 300 предложений (средняя длина — около 8 токенов или 60 символов) оценивается BLEU и ChrF. Базовый перевод gpt-4o-mini даёт BLEU 16.14 и ChrF 45.13; после правок через FST получили BLEU 25.71 и ChrF 50.87. Прирост составил +9.57 BLEU и +5.74 ChrF. Более резкий рост BLEU ожидаем: исправление одной именной группы зачастую «чинит» сразу несколько n-грамм подряд; ChrF реагирует умереннее, поскольку фиксирует локальные символные изменения (сводные результаты представлены на рис. 3).

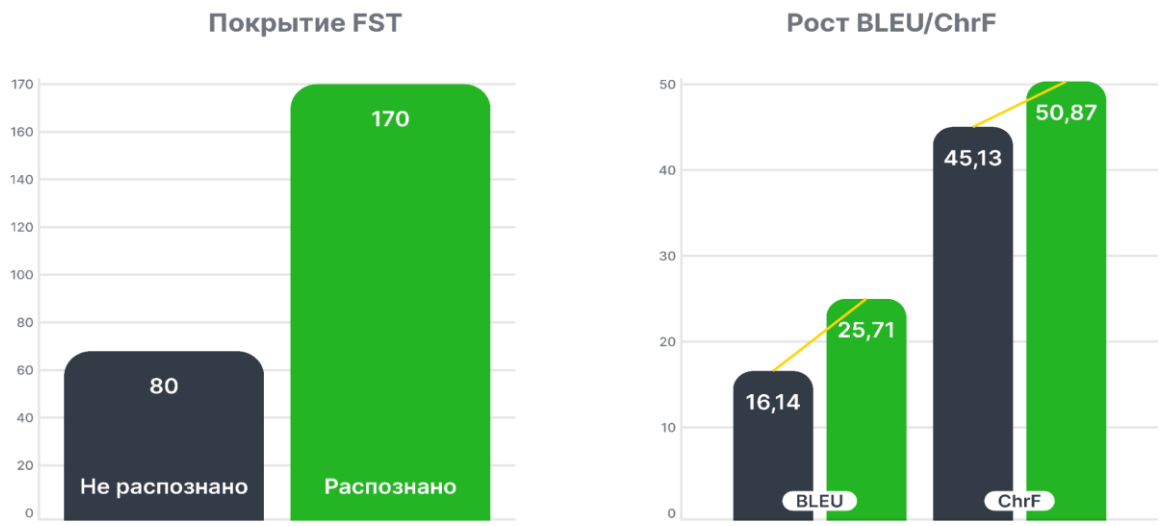

Рис. 3. Слева — покрытие FST по 250 словоформам; справа — прирост BLEU/ChrF после валидации через FST на 300 предложениях.

Важно, что gpt-4o-mini сам по себе относительно неплохо справляется с башкирским; на более слабых моделях выигрыш от данного подхода, вероятно, был бы еще заметнее. Поиск оптимальной базовой LLM находится за рамками настоящей работы, но сама архитектура к этому приспособлена: в гибридной схеме ключевая задача LLM состоит в том, чтобы извлекать лемму и теги, а не генерировать точную поверхность. Это означает, что компонент LLM можно заменять на более легкий/дешевый, обученный хуже на башкирском, FST всё равно возьмет на себя корректную реализацию поверхностной формы.

ЗАКЛЮЧЕНИЕ

Представлен гибридный нейросимволический подход к морфологически корректной генерации на агглютинативных языках: LLM извлекает из контекста пару «лемма + теги», а FST реализует и валидирует поверхностную форму. В автоматизированном режиме из корпусных данных извлекаются переносимые LEXC/regex-описания выбора алломорфов и фонологических чередований, что позволяет FST обобщаться на леммы вне исходного списка. Интеграция FST прямо в контур генерации дает заметный прирост качества (BLEU/ChrF) без дообучения переводчика, что подтверждает практичность подхода при ограниченных ресурсах.

Метод применим к задачам перевода, проверки орфографии и автоматизации процессов создания цифровых лингвистических ресурсов для малоресурсных языков, что особенно актуально для проектов, таких как «Тюркская морфема», где требуется предварительное заполнение грамматических шаблонов.

В будущем планируется расширять покрытие на другие морфологические категории (включая глаголы и другие части речи) и углублять автоматизацию индукции и верификации правил.

СПИСОК ЛИТЕРАТУРЫ

1. *Sproat R., Østling R.* The morphological gap between translation quality and surface accuracy // Proceedings of the WMT 2020 Conference. Online, 2020. P. 1015–1024.

2. *Kann K., Cotterell R., Schütze H.* Neural models of inflectional morphology // Proceedings of the 15th Conference of the European Chapter of the ACL (EACL 2017). Valencia, 2017. P. 322–334.

3. *Mielke S., Eisenstein J., Cotterell R.* Dialect-to-dialect translation and cross-dialect morphological robustness of language models // Transactions of the ACL. 2021. Vol. 9. P. 288–302.

4. *Koskenniemi K.* Two-level morphology: a general computational model for word-form recognition and production. Helsinki: University of Helsinki, Department of General Linguistics, 1983. 38 p.

5. *Beesley K.R., Karttunen L.* Finite-State Morphology. Stanford (CA): CSLI Publications, 2003. 550 p.

6. *Stahlberg F., Hasler E., Waite A.* SGNMT: A flexible NMT decoding toolkit for quick prototyping of new models // Proceedings of ACL System Demonstrations. Vancouver, 2017. P. 67–72.

7. *Hulden M.* FST-based grammar correction for richly inflected languages // Proceedings of ACL Workshop on Finite-State Methods. Montréal, 2012. P. 32–39.

8. *Tamchyna A., Bojar O.* Target-side context for morphological reinflection // Proceedings of the First Conference on Machine Translation (WMT 2016). Berlin, 2016. P. 586–594.

9. *Schwartz L., Liu S., Surrain S.* Bootstrapping a neural morphological analyzer from an existing FST // Proceedings of the ACL Workshop on Morphological Resources 2022. Seattle, 2022. P. 12–20.

NEURO-SYMBOLIC APPROACH TO AUGMENTED TEXT GENERATION VIA AUTOMATED INDUCTION OF MORPHOTACTIC RULES

M. V. Isangulov[1] [0009-0006-3244-0328], A. M. Elizarov[2] [0000-0003-2546-6897], A. R. Kunafin[3] [0009-0006-0495-265X], A. R. Gatiatullin[4] [0000-0003-3063-8147], N. A. Prokopyev[5] [0000-0003-0066-7465]

[1, 2]Kazan Federal University, Kazan, Russia

[3]Independent researcher

[4, 5]Academy of Sciences of the Republic of Tatarstan, Kazan, Russia

[1]marathon.our@gmail.com, [2]amelizarov@gmail.com, [3]aigizk@gmail.com

[4]ayrat.gatiatullin@gmail.com, [5]nikolai.prokopyev@gmail.com

Abstract

The work presents a hybrid neuro-symbolic method that combines a large language model (LLM) and a finite-state transducer (FST) to ensure morphological correctness in text generation for agglutinative languages. The system automatically extracts rules from corpus data: for local examples of word forms, the LLM produces sequences of morphological analyses, which are then aggregated and organized into compact descriptions of morphotactic rules (LEXC) and allomorph selection (regex). During generation, the LLM and FST operate jointly: if a token is not recognized by the automaton, the LLM derives a “lemma+tags” pair from the context, and the FST produces the correct surface form. A literary corpus (~1600 sentences) was used as the dataset. For a list of 50 nouns, 250 word forms were extracted. Using the proposed algorithm, the LLM generated 110 context-sensitive regex rules along with LEXC morphotactics, from which an FST was compiled that recognized 170/250 forms (~70%). In an applied machine translation test on a subcorpus of 300 sentences, integrating this FST into the LLM cycle improved quality from BLEU 16.14 / ChrF 45.13

to BLEU 25.71 / ChrF 50.87 without retraining the translator. The approach scales to other parts of speech (verbs, adjectives, etc.) as well as to other agglutinative and low-resource languages, where it can accelerate the development of lexical and grammatical resources.

***Keywords**: neuro-symbolic approach, large language model, finite-state transducers, two-level morphology, LEXC morphotactics, machine translation, agglutinative languages, Bashkir language.*

REFERENCES

1. *Sproat R., Østling R.* The morphological gap between translation quality and surface accuracy // Proceedings of the WMT 2020 Conference. Online, 2020. P. 1015–1024.

2. *Kann K., Cotterell R., Schütze H.* Neural models of inflectional morphology // Proceedings of the 15th Conference of the European Chapter of the ACL (EACL 2017). Valencia, 2017. P. 322–334.

3. *Mielke S., Eisenstein J., Cotterell R.* Dialect-to-dialect translation and cross-dialect morphological robustness of language models // Transactions of the ACL. 2021. Vol. 9. P. 288–302.

4. *Koskenniemi K.* Two-level morphology: a general computational model for word-form recognition and production. Helsinki: University of Helsinki, Department of General Linguistics, 1983. 38 p.

5. *Beesley K.R., Karttunen L.* Finite-State Morphology. Stanford (CA): CSLI Publications, 2003. 550 p.

6. *Stahlberg F., Hasler E., Waite A.* SGNMT: A flexible NMT decoding toolkit for quick prototyping of new models // Proceedings of ACL System Demonstrations. Vancouver, 2017. P. 67–72.

7. *Hulden M.* FST-based grammar correction for richly inflected languages // Proceedings of ACL Workshop on Finite-State Methods. Montréal, 2012. P. 32–39.

8. *Tamchyna A., Bojar O.* Target-side context for morphological reinflection // Proceedings of the First Conference on Machine Translation (WMT 2016). Berlin, 2016. P. 586–594.

9. *Schwartz L., Liu S., Surrain S.* Bootstrapping a neural morphological analyzer from an existing FST // Proceedings of the ACL Workshop on Morphological Resources 2022. Seattle, 2022. P. 12–20.

СВЕДЕНИЯ ОБ АВТОРАХ

ИСАНГУЛОВ Марат Вильданович окончил бакалавриат Института информационных технологий и интеллектуальных систем (ИТИС) Казанского (Приволжского) федерального университета в 2021 году, магистратуру ИТИС в 2023 г. В настоящее время – аспирант ИТИС.

Marat Vildanovich ISANGULOV graduated with a Bachelor's degree from Institute of Information Technology and Intelligent Systems (ITIS) of Kazan Federal University in 2021 and a Master's degree from ITIS in 2023. He is currently a PhD student at ITIS.

email: marathon.our@gmail.com

ORCID: 0009-0006-3244-0328

ЕЛИЗАРОВ Александр Михайлович – доктор физико-математических наук, профессор, заслуженный деятель науки Российской федерации, заслуженный деятель науки Республики Татарстан, профессор кафедры цифровой аналитики и технологий искусственного интеллекта Института информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета.

ELIZAROV Alexander Mikhailovich – Doctor of Physics and Mathematics, Professor, Honoured Worker of Science of the Russia, Honoured Worker of Science of the Republic of Tatarstan, Kazan Federal University. Current scientific interests: data mining, recommender systems, cloud computing, knowledge extraction technologies.

email: amelizarov@gmail.com

ORCID: 0000-0003-2546-6897

КУНАФИН Айгиз Ражапович. Окончил Уфимский авиационный университет. Создатель башкироязычной умной колонки «Һомай» и одного из крупнейших открытых ресурсов для цифровизации башкирского языка. Победитель конкурса AI for Good Innovation Factory Malta (2025); представлял проект «Homai» на финале в Женеве (предфинал, 2025). Соавтор публикации WMT-2021 по машинному переводу тюркских языков, контрибьютор Apertium (Bashkir), эксперт программы ЮНЕСКО «Информация для всех» с 2022 года.

Aygiz Razhapovich KUNAFIN graduated from Ufa Aviation University. Creator of the Bashkir-language smart speaker “Һомай” and a major open-source platform for digitizing the Bashkir language. He is the winner of the AI for Good Innovation Factory Malta pitch competition in March 2025; represented the project “Homai” at the Geneva Grand Finale of the AI for Good Global Summit in July 2025 (pre-final round). He is a co-author of a WMT-2021 publication on machine translation for Turkic languages, contributor to Apertium (Bashkir); invited expert in UNESCO’s “Information for All” Programme since 2022.

email: aigizk@gmail.com
ORCID: 0009-0006-0495-265X

ГАТИАТУЛЛИН Айрат Рафизович. Окончил Казанский государственный университет в 1994 г., к. т. н. (2002). Ведущий научный сотрудник Института прикладной семиотики Академии наук Республики Татарстан. Автор более 60 научных трудов.

Ayrat Rafizovich GATIATULLIN *g*raduated from Kazan State University in 1994, candidate in technical sciences (2002). He is a leading researcher at the Institute of Applied Semiotics of the Academy of Sciences of the Republic of Tatarstan. List of scientific works includes more than 60 publications.

email: ayrat.gatiatullin@gmail.com,
ORCID: 0000-0003-3063-8147

ПРОКОПЬЕВ Николай Аркадиевич. Кандидат технических наук. Окончил Институт вычислительной математики и информационных технологий Казанского федерального университета в 2015 году. Научный сотрудник Института прикладной семиотики Академии наук РТ. В списке научных трудов более 50 работ.

Nikolai Arkadievich PROKOPYEV candidate of Technical sciences. Graduated from the Institute of Computational Mathematics and Information Technologies of the Kazan Federal University in 2015. He is a researcher at the Institute of Applied Semiotics of the Academy of Sciences of the Republic of Tatarstan. List of scientific works includes more than 50 publications.

email: nikolai.prokopyev@gmail.com,

ORCID: 0000-0003-0066-7465

Материал поступил в редакцию 13 октября 2025 года