Построение эмбеддингов крымскотатарских слов: корпус, модели, валидация

Пеганов Никита

июнь 2025

Аннотация

В работе описан процесс построения корпуса текстов на крымскотатарском языке, очистки, лемматизации и создания семантических эмбеддингов (CBOW и SVD). Ключевая особенность — подробная валидация каждого шага с носителями языка с целью минимизации шума и смешанных слов. Отдельное внимание уделено успешному применению локальной LLM DeepSeek для фильтрации лексики. Ссылка на репозиторий: `https://github.com/NikPeg/qt-embeddings`

1 Введение

Обработка текстов на языках с низкими ресурсами (например, крымскотатарском) сталкивается с отсутствием инструментов, нечистыми корпусами, смешанным письмом и нехваткой лингвистических ресурсов. Семантические эмбеддинги для такого языка крайне важны: они позволяют автоматизировать задачи классификации, морфоанализа, машинного перевода, а также лингвистических исследований языков народов России.

Особенность нашего проекта — на каждом этапе обработки данные проверялись носителями крымскотатарского языка, чтобы исключить шум, ошибки, заимствования и обеспечить чистоту корпуса. Мы также применили современные нейросетевые инструменты (LLM DeepSeek) — и их результаты дополнительно валидировались носителями.

В перспективе планируется существенно увеличить корпус за счёт автоматического сбора (краулинга) с большего числа сайтов и ресурсов.

1.1 Команда

Пеганов Никита — единственный участник команды, выполнил сбор, обработку данных, построение моделей и написание отчёта. Все этапы предварительно консультировались с носителями языка.

2 Связанные работы

Классические эмбеддинги (Word2Vec CBOW, Skip-Gram [Mikolov et al., 2013], FastText [Bojanowski et al., 2017]) хорошо работают для ресурсных языков, но не применяются напрямую к крымскотатарскому из-за морфологических и орфографических особенностей, шума и смешения с другими языками.

Турецкие лемматизаторы, такие как Zeyrek [Akin and Akin, 2007] и Turkish-Lemmatizer [Koksal, 2015], были протестированы на крымскотатарских данных, однако показали низкое качество — высокая доля ошибок возникает из-за существенных различий между крымскотатарским и турецким языками, включая различие в суффиксах, морфологических правилах и словарном запасе.

Применение локальных LLM (например, DeepSeek [Team, 2024]) с несколькими вариациями промпта дало высокую точность фильтрации лексики, однако результаты требуют обязательной ручной валидации носителями языка.

Ранние успешные эксперименты по построению эмбеддингов для малых языков существуют, например, [Grave et al., 2018], где была показана возможность построения векторных представлений для многих слаборесурсных языков, используя в том числе данные Википедии и методы обогащения словарей.

3 Описание модели

Конвейер обработки представлен на рис. 1 и включает следующие этапы:

1. Сбор данных: Открытые крымскотатарские тексты из интернета, книг, СМИ.

2. Очистка и токенизация: Удаление мета-текста, нормализация орфографии и пунктуации, фильтрация русских/турецких и других фрагментов, разбиение на предложения и токены.

3. Лемматизация: Собственный гибридный лемматизатор: словарь + эвристики для морфологии, сравнение с турецким и LLM-инструментами, ручное подтверждение чистоты.

4. Минимизация шума: На каждом этапе результаты сверялись с носителями языка — это многократно уменьшило количество "мусорных" и лишних слов.

5. Применение LLM (DeepSeek): Был развёрнут локальный LLM DeepSeek-R1-Distill-Qwen-32B, к которой применялись различные промпты для определения, является ли слово крымскотатарским. Классификация неоднозначных лексем проводилась с несколькими проходами; итоговые списки подтверждал носитель.

6. Построение эмбеддингов: Тренировка эмбеддингов CBOW (gensim), дополнительно — SVD по матрице совместных появлений.

Рис. 1: Схема обработки корпуса и построения эмбеддингов.

4 Корпус данных

Датасет для обучения был собран на основе открытых источников: крымскотатарских сайтов, электронных книг, газет и архивов СМИ. Детализированная статистика по количеству документов, предложений и токенов приведена в таблице 1.

Для автоматизации сбора был разработан и использован отдельный пакет краулеров, исходный код которого доступен по ссылке: `https://github.com/NikPeg/qirimtatar-embedding-crawlers`. С помощью этого инструмента были загружены и предобработаны данные из различных интернет-источников — от литературы до современных публикаций. Краулеры поддерживают не только парсинг html-страниц, но и обработку документов в форматах DOC, PDF, а также распознавание текстов с изображений при помощи Tesseract OCR. Особое внимание уделялось корректному определению крымскотатарского языка (через Яндекс API) и фильтрации нерелевантных материалов (например, обучение, материалы на русском языке и пр.).

Парсер содержит многопоточные модули для быстрой выгрузки архивов, систему удаления нежелательных водяных знаков (Aspose), а также полноценный конвертер изображений и PDF в текстовый формат с учётом особенностей обработки крымскотатарского письма.

Процесс сбора и предобработки включал:

- автоматизированный сбор текстов с различных источников (сайты, гугл-документы, архивы и пр.);
- извлечение текста из PDF, DOC и изображений (OCR);
- фильтрацию языковых артефактов и удаление мусорных файлов;
- определение языка каждого фрагмента с помощью Яндекс API;
- удаление учебных текстов и материалов на других языках на основе стоп-слов и ручной проверки.

Итоговые наборы документов (более 10 ГБ в сжатом виде) доступны для скачивания по ссылкам, приведённым в README краулер-репозитория [Peganov, 2024].

Очищаются:

- Русские, турецкие и любые не-qt-фрагменты
- Нормализация орфографии, пунктуации
- Разметка предложений, токенизация
- Лемматизация и фильтрация — всё валидировано носителями языка

Обработано ∼309 тыс. предложений и более 3 млн токенов, извлечено 8245 лемм.

	Значение
Файлов	123
Предложений	309,334
Токенов	3,157,286
Лемм	8,245
Оценка OoV	<5% (после очистки)

Таблица 1: Статистика корпуса крымскотатарских текстов. После финальной фильтрации доля "мусора" минимальна.

Будущая работа: в перспективе планируется существенное увеличение корпуса за счёт автоматического сбора текстов с сайтов и форумов.

5 Эксперименты

5.1 Метрики

Для оценки применялись:

- Доля корректно лемматизированных и распознанных токенов (по оценке носителей)

- Семантическая связность соседей в эмбеддингах (ручная проверка)
- Сравнение OoV после разных фильтров: Zeyrek, DeepSeek и примитивный baseline

5.2 Экспериментальная настройка

Проводилось:

- Сравнение разных подходов к лемматизации/фильтрации: свой, Zeyrek, DeepSeek
- Тренировка CBOW-эмбеддингов (50 эпох, размер вектора 100, окно 5, мин.частота 3), SVD-эмбеддингов (100 компонент)
- Весь результат этапов валидировался носителями языка — особое внимание к самым частотным словам

5.3 Бейслайны

Использовались:

- Примитивная лемматизация и фильтрация
- Турецкий лемматизатор Zeyrek [Akin and Akin, 2007]
- Турецкий лемматизатор Turkish-Lemmatizer [Koksal, 2015]
- LLM DeepSeek для языковой фильтрации
- Случайный/частотный baseline для анализа ошибок

6 Результаты

Чистота корпуса: Валидация на всех этапах сильно снизила шум и количество не-крымскотатарских слов. Правила+словари дали покрытие 25.5% по леммам, Zeyrek — только 2.8%. Осталось ≈14 тыс. нерспознанных токенов (vs 27 тыс. у Zeyrek).

Собственный лемматизатор: В процессе построения корпуса был разработан специализированный лемматизатор, основанный на расширенной версии турецкого лемматизатора Turkish-Lemmatizer с добавлением крымскотатарских словарей, правил и морфологических шаблонов. Данная система показала значительно лучшие результаты по сравнению с турецкой версией и может быть использована как самостоятельный инструмент для обработки текстов на крымскотатарском языке в будущих исследовательских и прикладных задачах.

LLM-фильтрация: Локальный DeepSeek с несколькими итерациями промпта показал отличную точность в фильтрации не-qt-лексики; все сомнительные случаи рассматривались с носителем.

Эмбеддинги: CBOW и SVD хорошо группируют семантически близкие слова (см. табл. 2).

Запрос	**1**	**2**	**3**	**4**
`халкъ` (народ)	adam (человек)	дюль (сердце/разум)[1]	adalarında (на островах)	джеси (его/её часть)[2]
`джан` (душа)	ешерди (прятал(а))	гоньдже (юная девушка)	узьди (оторвал(а))	сёзлерининъ (его/её слова)
`мектеп` (школа)	мудири (директор)	бетине (на лицо)	эмизе (мать)	taze (новый, свежий)
`миллет` (нация)	атар (племя)[3]	пери (фея)	голланд (Голландия)	тенде (на теле)
`тиль` (язык)	давушнен (с голосом)	менимкини (мой)	суреттс (изобрази)	тюшюндим (я понял)

Таблица 2: Ближайшие соседи к частотным словам (CBOW).

Качественный вывод: Семантика сохраняется: например, соседи слова "халкъ" (народ) — слова о сообществе, у "агъыз" (рот) — части тела и речь.

7 Выводы

Построен чистый валидированный корпус крымскотатарских текстов, леммы и эмбеддинги (CBOW, SVD). Все этапы были проверены носителями языка — это позволило значительно сократить шум, уменьшить смешение с другими языками. Локально применён LLM DeepSeek для языкового фильтра с несколькими раундами промпта — и подтверждён носителем как рабочий инструмент.

Корпус, эмбеддинги и лемматизированная лексика пригодны для дальнейших NLP, типологических и лингвистических задач для низкоресурсных языков. В перспективе — расширение корпуса за счёт краулинга, проверка эмбеддингов на реальных downstream-задачах, тиражирование инструментов на другие языки народов России.

Список литературы

[Akin and Akin, 2007] Akin, A. and Akin, M. (2007). Zemberek-nlp: An open source nlp framework for turkic languages.

[1] Значение слова "дюль" требует уточнения, возможно, это форма слова "диля" (сердце, ум).

[2] Форма окончания, значение зависит от контекста.

[3] В тюркских языках "атар" — "род", "племя" или форма глагола "стрелять".

[Bojanowski et al., 2017] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

[Grave et al., 2018] Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.

[Koksal, 2015] Koksal, A. (2015). Turkish-lemmatizer. `https://github.com/akoksal/Turkish-Lemmatizer`.

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

[Peganov, 2024] Peganov, N. (2024). Qirimtatar embedding crawlers. `https://github.com/NikPeg/qirimtatar-embedding-crawlers`.

[Team, 2024] Team, D. (2024). Deepseek r1 distill qwen-32b. `https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-32B`.