DOI 10.37882/2223-2966.2023.07.01

ЛЕММАТИЗАЦИЯ СУЩЕСТВИТЕЛЬНЫХ В АЗЕРБАЙДЖАНСКОМ ЯЗЫКЕ

LEMMATIZATION OF NOUNS IN AZERBAIJANI LANGUAGE

A. Aghayev
S. Molodyakov

Summary: This paper presents a rule-based and regular expression-based lemmatization algorithm for Azerbaijani language nouns. The algorithm's effectiveness was evaluated using performance and accuracy tests. The results demonstrated high accuracy and efficiency of the developed algorithm, making it attractive for use in natural language processing systems for the Azerbaijani language.

Keywords: lemmatization, Azerbaijani language, nouns, regular expressions, natural language processing, morphological analysis.

Агаев Аслан Фахри оглы
Аспирант, Санкт-Петербургский политехнический университет Петра Великого
agaev.af@edu.spbstu.ru
Молодяков Сергей Александрович
д.т.н., профессор, Санкт-Петербургский политехнический университет Петра Великого
molodyakov_sa@spbstu.ru

Аннотация. В данной работе представлен алгоритм лемматизации существительных азербайджанского языка на основе правил и регулярных выражений. Эффективность алгоритма была оценена с использованием тестов на производительность и точность. Результаты показали высокую точность и эффективность разработанного алгоритма, что делает его привлекательным для использования в системах обработки естественного азербайджанского языка.

Ключевые слова: лемматизация, азербайджанский язык, существительные, регулярные выражения, обработка естественного языка, морфологический анализ.

Введение

Обработка естественного языка (Natural Language Processing, NLP) является разделом искусственного интеллекта, который изучает взаимодействие между компьютерами и человеческим языком с целью автоматического понимания, интерпретации и генерации текста. NLP использует множество алгоритмов и методов для анализа текста, извлечения информации, автоматического перевода и других задач, связанных с обработкой текстов на различных языках.

Большие языковые модели, такие как LLaMA [1], являются высокоэффективными и популярными инструментами в области NLP. Они обучены на огромных объемах данных и способны выполнять широкий спектр задач, связанных с обработкой и генерацией текста. Благодаря своей мощности и универсальности, такие модели существенно повышают качество обработки текста и предоставляют новые возможности для разработки приложений на основе языка.

Однако, несмотря на успехи больших языковых моделей, базовые инструменты NLP для конкретных языков, особенно менее популярных, по-прежнему актуальны. Среди причин их актуальности можно выделить:

— Лингвистическое разнообразие: создание инструментов NLP для разных языков способствует лингвистическому разнообразию и помогает, чтобы менее популярные языки не оставались в стороне в эпоху цифровых технологий.
— Научные исследования и инновации: работа над менее популярными языками может привести к интересным научным вызовам и инновациям в области NLP.
— Расширение доступности искусственного интеллекта: поскольку приложения на основе искусственного интеллекта и NLP продолжают расширяться, обеспечение доступности таких технологий для носителей менее популярных языков может содействовать созданию более инклюзивных систем на основе искусственного интеллекта.

Таким образом, противопоставляя базовые инструменты NLP большим моделям, в пользу первых можно сказать следующее: большие модели требуют большого количества ресурсов, которые не всегда доступны для менее популярных языков; большие модели сложнее использовать для выполнения узкоспециализированных задач в конкретной доменной области; базовые инструменты NLP часто являются необходимыми элементами при создании больших языковых моделей.

Исходя из вышеперечисленных соображений, можно с уверенностью заявить, что создание лемматизатора для азербайджанского языка является актуальной и важной задачей.

Одним из базовых инструментов NLP является лемматизация [2]. Лемматизация — это процесс преобразования слова в его базовую форму или лемму. Этот инструмент играет важную роль в анализе текста, упрощая

процесс обработки и позволяя компьютерам группировать различные формы одного слова для более точного анализа контекста и извлечения информации.

В данной статье мы описываем алгоритм лемматизации для азербайджанского языка. Возможные применения лемматизации включают информационный поиск, машинный перевод, анализ тональности текста, извлечение информации и другие задачи NLP. Разработка такого инструмента имеет особую актуальность в свете недостатка специализированных ресурсов и инструментов для азербайджанского языка в сравнении с более широко распространенными языками.

Алгоритм лемматизации, представленный в данной работе, основывается на морфологическом анализе азербайджанского языка и учете его особенностей. Мы исследуем различные подходы к лемматизации, сравниваем их эффективность и предлагаем оптимальный подход для азербайджанского языка. В процессе разработки алгоритма мы также учитываем особенности морфологии азербайджанского языка.

В нашем исследовании мы также анализируем существующие корпусы азербайджанского языка, ресурсы и инструменты, доступные для работы с ним, и определяем потребности в дополнительных ресурсах и улучшении существующих инструментов. Мы проводим эксперименты, чтобы оценить производительность предложенного алгоритма лемматизации и сравнить его с альтернативными подходами и существующими инструментами.

Обзор существующих решений

Существует несколько подходов к лемматизации, и в этой главе рассматриваются три распространенных метода: основанные на правилах, основанные на поиске и основанные на машинном обучении.

Алгоритмы основанной на правилах лемматизации основаны на наборе предопределенных лингвистических правил и морфологических преобразований для определения леммы слова. Эти правила, как правило, разрабатываются лингвистическими экспертами и могут включать удаление аффиксов, преобразование неправильных глаголов или изменение множественных форм на единственное число. Основанные на правилах методы зависят от языка и требуют глубокого понимания морфологии и грамматики целевого языка.

Известный основанный на правилах лемматизатор — это алгоритм Портера, разработанный Мартином Портером в 1980 году. Хотя это в основном алгоритм стемминга, он работает на основе основанного на правилах принципа, применяя преобразования к словам для их сведения к базовой форме [3].

Лемматизация на основе поиска использует предварительно созданный словарь или лексикон, содержащий словоформы и соответствующие им леммы. Получив на вход слово, алгоритм ищет его в лексиконе и возвращает связанную лемму. Этот подход подходит для языков с ограниченным числом неправильных форм или языков с простой морфологией. Однако он может не сработать хорошо для сильно флективных языков, так как размер лексикона может стать громоздким.

Лемматизатор WordNet, являющийся частью библиотеки Natural Language Toolkit (NLTK) для Python, является пример ом лемматизатора на основе поиска. Он основан на лексической базе данных WordNet для выполнения лемматизации [4].

Алгоритмы лемматизации на основе машинного обучения используют методы обучения с учителем или без учителя для определения леммы слова. Эти методы могут автоматически изучать сложные морфологические правила из обучающих данных, делая их более адаптивными к новым языкам и языковым вариациям. Методы с учителем требуют размеченных обучающих данных, в то время как методы без учителя могут изучать закономерности из неразмеченного текста.

Выдающимся примером лемматизатора на основе машинного обучения является нейронная модель последовательность-к-последовательности [5]. Этот подход заключается в обучении нейронной сети предсказывать лемму данной словоформы путем преобразования ее в последовательность символов и изучения отображения между входными и выходными последовательностями.

Основанные на правилах подходы: основанные на правилах методы могут быть сложными в разработке, особенно для языков с богатой морфологией и неправильными формами. Они также могут потребовать значительного времени и усилий для поддержания и обновления правил и словарей, чтобы соответствовать изменениям языка.

Подходы на основе поиска: эти методы зависят от наличия полного и точного словаря, что может быть проблемой для малоизученных языков. Кроме того, поиск по словарю может быть медленным и неэффективным для больших объемов данных.

Методы на основе машинного обучения: модели машинного обучения могут быть сложными в настройке и требовать больших вычислительных ресурсов. Они также могут потребовать больших размеченных обучающих данных, что может быть проблемой для малоизученных языков.

В случае азербайджанского языка, основанный на правилах подход может быть предпочтительным по следующим причинам:

Контролируемость: основанные на правилах методы позволяют разработчикам иметь полный контроль над процессом лемматизации, что может облегчить отладку и устранение ошибок.

Прозрачность: правила и словари могут быть легко понятыми и анализируемыми, что делает подход более интерпретируемым и объяснимым.

Отсутствие необходимости в больших обучающих данных: основанные на правилах методы не требуют размеченных обучающих данных, что может быть особенно полезно для малоизученных языков, где такие данные могут быть недоступны или ограничены.

Низкие вычислительные требования: основанные на правилах методы обычно требуют меньше вычислительных ресурсов, чем подходы на основе машинного обучения, что делает их более доступными для широкого круга пользователей и приложений.

Выбор основанного на правилах подхода для азербайджанского языка может быть обусловлен его преимуществами в контролируемости, прозрачности и доступности. Однако стоит отметить, что эффективность основанного на правилах подхода может зависеть от качества разработанных правил и словарей, а также от особенностей морфологии азербайджанского языка.

Тем не менее, основанный на правилах подход может быть хорошим началом для работы с азербайджанским языком, особенно если нет доступа к большим размеченным данным или достаточным вычислительным ресурсам для использования методов машинного обучения. В долгосрочной перспективе, с развитием технологий и наращиванием доступных данных, можно рассмотреть возможность интеграции других подходов к лемматизации. Это может помочь дополнить и улучшить основанные на правилах методы, особенно в сложных и неоднозначных случаях, и сделать лемматизацию более гибкой и адаптивной.

Один из сравнительно новых основанных на правилах подходов к лемматизации для турецкого языка был представлен в статье «A Morphology-based Turkish Text Lemmatizer» в 2016 году [6] . Этот подход использует морфологические правила, основанные на агглютинативной структуре турецкого языка, и был разработан с использованием открытого морфологического анализатора для турецкого языка под названием Zemberek.

В этом подходе лемматизация осуществляется в два этапа:

Морфологический анализ: Текст анализируется с помощью Zemberek, который разбивает слова на морфемы и определяет их морфологические характеристики, такие как время, род и падеж.

Применение правил лемматизации: на основе полученной информации разрабатываются правила лемматизации, которые позволяют определить основу слова и сопоставить ее с соответствующей леммой. Эти правила учитывают морфологические свойства слова и агглютинативную структуру турецкого языка.

Авторы статьи заявляют, что этот подход показывает хорошие результаты для турецкого языка, достигая точности лемматизации более 95 %. Они также отмечают, что метод может быть дополнен другими подходами, такими как машинное обучение, для дальнейшего улучшения качества лемматизации [7].

Предлагаемый подход

В данной статье предлагается использовать подход на основе регулярных выражений для реализации алгоритма лемматизации азербайджанских существительных. Подход на основе регулярных выражений обладает рядом преимуществ по сравнению с другими методами, такими как конечные автоматы (состояния) или морфологические анализаторы [8].

Во-первых, регулярные выражения предоставляют компактный и четкий способ описания грамматических правил, что облегчает чтение и понимание кода. Это также упрощает добавление новых правил и корректировку существующих, что повышает гибкость алгоритма и его применимость к различным вариантам языка [10].

Во-вторых, регулярные выражения являются стандартным инструментом для работы со строками в большинстве языков программирования. Использование встроенных возможностей языка программирования обеспечивает оптимальную производительность и надежность алгоритма, снижая вероятность возникновения ошибок и упрощая процесс отладки [9].

Тем не менее, подход на основе регулярных выражений может быть менее наглядным и трудным для понимания для сложных грамматических структур по сравнению с подходами, используемыми конечные автоматы или морфологические анализаторы. Однако для задачи лемматизации азербайджанских существительных с представленным набором грамматических правил подход на основе регулярных выражений оказывается более простым и эффективным.

Таким образом, в данной статье предлагается использовать подход на основе регулярных выражений для

лемматизации азербайджанских существительных, так как он обеспечивает простоту, гибкость и высокую производительность.

Алгоритм лемматизации на основе правил и регулярных выражений

В рамках данной работы был разработан алгоритм лемматизации для азербайджанских существительных на основе правил и регулярных выражений. Основное преимущество данного подхода заключается в оптимальности решения, так как регулярные выражения позволяют эффективно работать со строками и обрабатывать сложные правила морфологии языка.

Алгоритм начинает работу с определения порядка суффиксов для существительных азербайджанского языка. Лемма (корень слова) может быть расширена суффиксами нескольких типов. Список возможных суффиксов приведен в табл. 1.

Таблица 1.

Список суффиксов

Тип	Список суффиксов
Принадлежность	‘ım’, ‘im’, ‘um’, ‘üm’, ‘m’, ‘ımız’, ‘imiz’, ‘umuz’, ‘ümüz’, ‘mız’, ‘miz’, ‘muz’, ‘müz’, ‘ın’, ‘ın’, ‘un’, ‘ün’, ‘n’, ‘ınız’, ‘iniz’, ‘unuz’, ‘ünüz’, ‘nız’, ‘niz’, ‘nuz’, ‘nüz’, ‘ı’, ‘i’, ‘u’, ‘ü’, ‘sı’, ‘si’, ‘su’, ‘sü’
Падеж	‘ın’, ‘un’, ‘ün’, ‘in’, ‘a’, ‘ə’, ‘ya’, ‘yə’, ‘i’, ‘ı’, ‘u’, ‘ü’, ‘da’, ‘də’, ‘dan’, ‘dən’
Число	‘lar’, ‘lər’
Соединители	‘y’, ‘n’, ‘s’

Важно отметить, что некоторые суффиксы могут следовать только за определенными суффиксами, а некоторые переходы не являются возможными. Это влияет на порядок и правила обработки суффиксов в алгоритме. Возможный порядок употребления суффиксов проиллюстрирован на рис. 1.

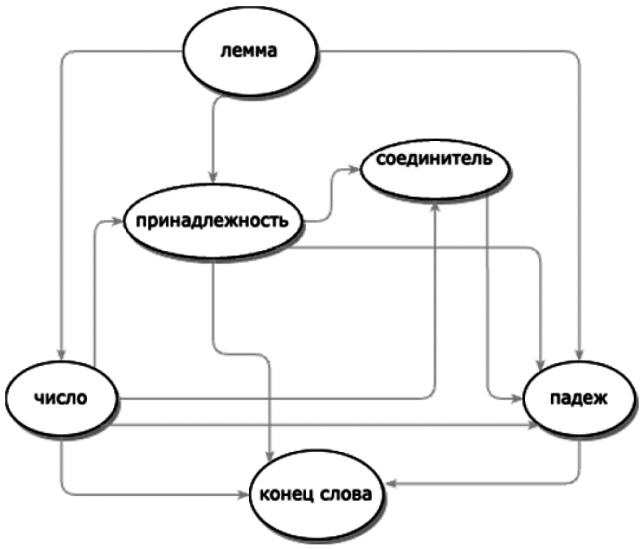

Рис. 1. Возможный порядок употребления суффиксов

Происходит поиск возможных цепочек суффиксов в соответствии с указанным порядком употребления. Путем отсечения цепочек суффиксов формируются возможные леммы и добавляются в список кандидатов.

Далее происходит ранжирование списка кандидатов.

Результатом применения алгоритма является список лемм с указанием степени достоверности. Основные этапы работы алгоритма приведены на рис. 2.

Рис. 2. Этапы работы алгоритма

Расчет степени достоверности возможных лемм происходит следующим образом:

— Если найдено больше одной возможной леммы, каждая из лемм получает степень достоверности $p = \frac{1}{n}$, где n — количество возможных лемм.

— Проверяется наличие каждой леммы в имеющемся словаре. Если лемма не найдена в словаре, ее степень достоверности уменьшается в два раза.

Далее список возможных лемм сортируется согласно степени достоверности и длине, то есть если две возможные леммы имеют одинаковую степень достоверности, приоритет будет отдан более длинной. Выходным параметром алгоритма является список лемм с указанием степеней достоверности.

Таким образом, разработанный алгоритм лемматизации на основе правил и регулярных выражений предоставляет эффективное и оптимальное решение для обработки азербайджанских существительных. Он учитывает морфологические особенности языка и обеспечивает корректное определение лемм слов с учетом порядка и возможных комбинаций суффиксов.

Результаты

Алгоритм был реализован на языке программирования Python 3.

Для проведения экспериментов и оценки алгоритма лемматизации использовался орфографический словарь азербайджанского языка 2021 года. Этот словарь содержит обширную коллекцию слов и является хорошим источником данных для анализа и проверки работы алгоритма.

Тестовые данные для измерения точности лемматизации были составлены на основе статей Википедии на азербайджанском языке, из которых были извлечены только существительные. Это позволило создать репрезентативный набор данных для оценки работы алгоритма. Для проведения экспериментов был использован набор из 1000 существительных.

Производительность алгоритма лемматизации была оценена путем измерения времени работы программы, реализующей алгоритм. Результаты тестов показали следующие значения:

Среднее время обработки одного слова: 3 мс.

Среднее количество обработанных слов в секунду: 333 слов/сек.

Однако следует отметить, что скорость работы алгоритма не может быть напрямую сравнена с результатами других исследований и методов лемматизации, так как факторы, влияющие на производительность, могут сильно различаться. К таким факторам относятся аппаратные характеристики, используемые программные библиотеки и даже реализация самого алгоритма.

Приведем пример лемматизации слова с помощью разработанной программы в табл. 2.

Таблица 2.

Пример обработки слова алгоритмом

Входное слово	Входное слово с указанием суффиксов	Найденная лемма
evlərindən (перевод: из их домов)	ev — lər (число) — in (принадлежность) — n (соединитель) — dən (падеж)	ev Степень достоверности: 1 (слово найдено в словаре)

Для оценки точности алгоритма лемматизации был проведен анализ правильности определения лемм на основе тестовых данных, составленных из существительных Википедии. Результаты эксперимента показали следующую точность:

Точность лемматизации составила 94 %.

Это свидетельствует о высокой точности и эффективности разработанного алгоритма лемматизации на основе правил и регулярных выражений для азербайджанского языка.

Заключение

В рамках данной работы была достигнута поставленная цель — разработан эффективный лемматизатор для существительных азербайджанского языка. В качестве подхода был выбран алгоритм лемматизации на основе правил и регулярных выражений, что оказалось оптимальным решением в текущей ситуации.

Полученные результаты оказались весьма обнадеживающими и близкими к идеальным, что свидетельствует о высокой эффективности разработанного алгоритма. Точность лемматизации составила 94 %, что является хорошим показателем для сложной морфологии азербайджанского языка.

Разработанный лемматизатор является важным шагом вперед в области обработки естественного языка для азербайджанского языка и создает основу для дальнейшего развития морфологического анализа азери. Возможные направления для дальнейшей работы включают расширение алгоритма для обработки других частей речи, улучшение точности лемматизации и интеграция с другими компонентами систем обработки естественного языка.

ЛИТЕРАТУРА

1. Touvron, Hugo et al. «LLaMA: Open and Efficient Foundation Language Models». ArXiv abs/2302.13971 (2023).
2. Hotho, A., Nürnberger, A., & Paaß, G. (2005). A Brief Survey of Text Mining. LDV Forum, 20(1), 19–62.
3. Porter, M.F. (1980). An algorithm for suffix stripping. Program, 14(3), 130–137
4. Fellbaum, C. (Ed.). (1998). WordNet: An Electronic Lexical Database. MIT Press.
5. Kann, K., & Schütze, H. (2016). Single-Model Encoder-Decoder with Explicit Morphosyntactic Decoding for Morphological Disambiguation. In Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers (COLING 2016), pp. 1051–1061.
6. Özçelik, R., & Eryiğit, G. (2016). A Morphology-based Turkish Text Lemmatizer. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), pp. 2811–2815.
7. Kondratyuk, D., & Straka, M. (2019). 75 Languages, 1 Model: Parsing Universal Dependencies Universally. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).
8. Köksal, A., & Özgür, A. (2016). A Morphology-based Turkish Text Lemmatizer. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16).
9. Jurafsky, D., & Martin, J.H. (2009). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.
10. Chapelle, C.A. (2012). The Encyclopedia of Applied Linguistics. Malden, MA: Wiley-Blackwell.

Журнал «Современная наука: актуальные проблемы теории и практики»