

МИНОБРНАУКИ РОССИИ САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Факультет КТИ

Кафедра АМ

Лемматизация текста

По дисциплине «Обработка естественных языков»

Работу выполнил:

Серафим Кириллович Иванов

Группа: 0385

Преподаватель:

Малютин Егор Владимирович

Санкт-Петербург

2025

Содержание

1. Постановка задачи	3
2. Ход работы	4
2.1. Идейное описание	4
2.2. Алгоритм лемматизации	5
2.3. Решение проблемы с омонимами	5
2.4. Качество работы алгоритма	5
3. Выводы	6

1. Постановка задачи

Требуется реализовать процедуру лемматизации текста: привести каждое слово к словарной форме и приписать соответствующий частеречный тег.

Для решения задачи допускается использовать данные, рассмотренные на лекциях (например, словарь oDict, разметку OpenCorpora и др.). Запрещается применять готовые морфологические анализаторы (mystem, pymorphy и т. п.).

Входные данные. Набор предложений вида

токен₁ токен₂ ...токен_N

с расставленными знаками препинания, разделённых переводом строки. В предложениях могут встречаться только следующие знаки препинания: запятая, точка, вопросительный и восклицательный знак.

Выходные данные. Для каждого предложения требуется вывести результат в виде:

токен₁{лемма₁=тег₁} токен₂{лемма₂=тег₂} ...токен_N{лемма_N=тег_N}

при этом исходные знаки препинания в выходе не отображаются. Разделителем между токенами служит пробел.

Замечания.

- При лемматизации буквы «е» и «ё», а также строчные и прописные варианты букв считаются равноправными.
- Для удобства анализа теги следует выбирать из предложенного в лекциях набора (S, V, A, ADV, PR, CONJ, NI и др.).

Пример.

Input:

Стала стабильнее экономическая и политическая обстановка, предприятия вывели из тени зарплаты сотрудников. Все Гришины одноклассники уже побывали за границей, он был чуть ли не единственным, кого не вывозили никуда дальше Красной Пахры.

Output:

Стала{стать=V} стабильнее{стабильный=A} экономическая{экономический=A} и{и=CONJ} политическая{политический=A} обстановка{обстановка=S} предприятия{предприятие=S} вывели{вывести=V} из{из=PR} тени{тень=S} зарплаты{зарплата=S} сотрудников{сотрудник=S} Все{весь=NI} Гришины{гришин=A} одноклассники{одноклассник=S} уже{уже=ADV} побывали{побывать=V} за{за=PR} границей{граница=S} он{он=NI} был{быть=V} чуть{чуть=ADV} ли{ли=ADV} не{не=ADV} единственным{единственный=A} кого{кто=NI} не{не=ADV} вывозили{вывозить=V} никуда{никуда=NI} дальше{далеко=ADV} Красной{красный=A} Пахры{Пахра=S}

2. Ход работы

Работа выполнена в Python с использованием формата ноутбука *.ipynb. Ознакомиться с работой можно в репозитории по приложенной [ссылке](#). Там же подробное описание всех действий.

В отчете будут краткие сведения о проделанной работе.

2.1. Идейное описание

Решено использовать словарь OpenCorpora, распространяющийся свободно [1]. Словарь представляет собой набор данных с морфологической информацией о словах русского языка, включающий леммы, формы слов и их частеречные характеристики. Он позволяет связывать каждое слово с его нормальной формой и набором грамматических признаков, что делает его удобным источником для решения задачи лемматизации.

В ходе выполнения лабораторной работы был разработан ноутбук на языке Python, реализующий задачу лемматизации текста с использованием словаря OpenCorpora.

Основные этапы работы включают:

- Подготовка данных.** Автоматическая загрузка словаря OpenCorpora и текстов («Война и мир», «Преступление и наказание») в рабочую директорию.
- Обработка словаря.** Распаковка и парсинг XML-файла, построение словаря в формате pickle, содержащего отображение словоформ в леммы и части речи.
- Лемматизация текста.** Разработаны функции для нормализации слов, лемматизации токенов и обработка предложений. Реализован вывод в формате токен{лемма=POS}.
- Проверка работы.** На примере небольшого текста продемонстрирован результат лемматизации, а также собрана статистика по количеству обработанных токенов и

корректности разметки.

2.2. Алгоритм лемматизации

Алгоритм лемматизации включает следующие шаги:

1. Токенизация входного текста.
2. Очистка токенов от знаков препинания.
3. Приведение токенов к нижнему регистру и замена буквы «ё» на «е».
4. Поиск токена в словаре: при успешном совпадении — извлечение леммы и части речи. При этом поиск выполняется за $\mathcal{O}(1)$ за счет того, что используется формат `dict` (то есть хэш таблица);
5. Формирование выходной строки в формате `токен{лемма=POS}`.

2.3. Решение проблемы с омонимами

При наивной реализации возникает проблема с омонимами: например, в предложениях "я уж не знаю, что тебе сказать" и "мой уж не съел лягушку" — слово "уж" относится к разным частям речи. В работе рассматривается следующее решение:

1. С помощью `rzmorphy3` построить словарь триграмм (всех возможных последовательностей из трех частей речи) и вычислить для них вероятности;
2. На основании словаря частот подбирать наиболее вероятную часть речи.

Такой вариант не даст идеального результата, но способен значительно улучшить точность результата.

2.4. Качество работы алгоритма

В ходе экспериментов процент нераспознанных слов оказался незначительным. Это свидетельствует о достаточной полноте словаря `OpenCorpora` и корректности реализованного алгоритма лемматизации. Большая часть токенов успешно приводится к нормальной форме с правильным определением части речи.

Небольшое количество ошибок связано в основном с:

1. Наличием редких имён собственных и заимствованных слов, отсутствующих в словаре;
2. Сокращениями и нестандартными формами, встречающимися в текстах.

Таким образом, предложенный метод можно считать надёжным для лемматизации большинства текстов на русском языке.

С другой стороны, замерять время работы нецелесообразно, поскольку даже обработка значительных текстов (книг Достоевского и Толстого) заняли меньше 0.3 секунды. Такой результат получен за счет преобработки словаря: из XML файла были удалены ненужные для работы данные, а затем он был конвертирован в словарь (хэш таблицу), а значит поиск имеет константную сложность. Таким образом, единственное слабое место — преобработка, которая занимает порядка минуты.

3. Выводы

Выполнена лемматизация текста на русском языке без использования готовых морфологических анализаторов. Алгоритм показал высокую точность: большинство слов успешно распознано, доля нераспознанных токенов мала (менее 5% для рассмотренной литературы). Использование словаря OpenCorpora и хэш-таблицы обеспечивает быстрый поиск и стабильную работу на больших текстах. Основное время затрачивается на предобработку словаря (около минуты на весь словарь OpenCorpora); последующая обработка текстов происходит практически мгновенно (менее 0.3 секунды на 460 000 слов).

Список литературы

- [1] Словарь / OpenCopora. — URL: <https://opencorpora.org/dict.php> (дата обращения: 17 сентября 2025).